

# Research Skills

## *Figures & Visuals*

Andrea Bajcsy



Join our Discord to test V3!

**Craiyon** V2

Free online AI image generator from text

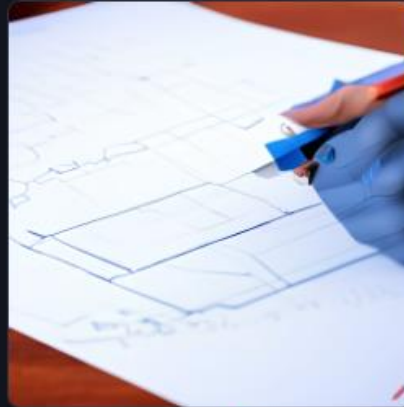


Draw

an academic figure for a computer vision research paper



 Draw




an academic graph for a computer vision paper

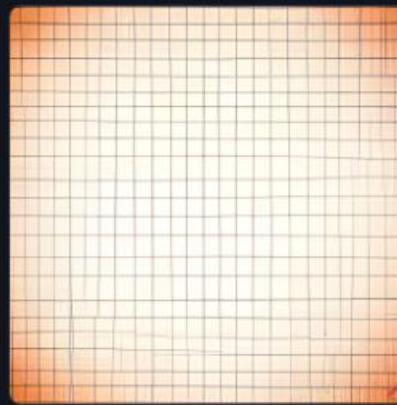
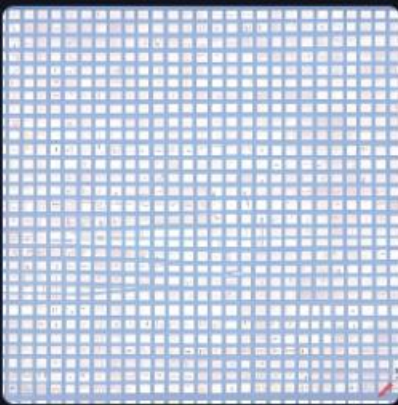
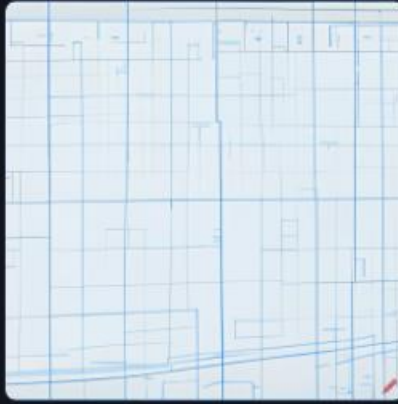
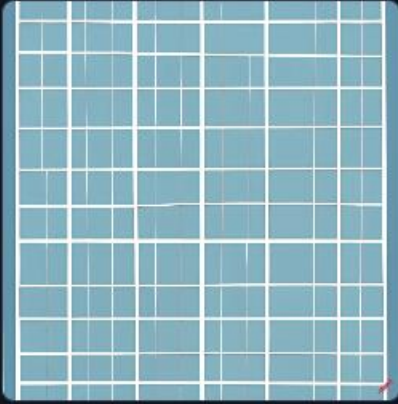


Draw

an academic graph for a computer vision paper



 Draw



scatterplot of sine wave

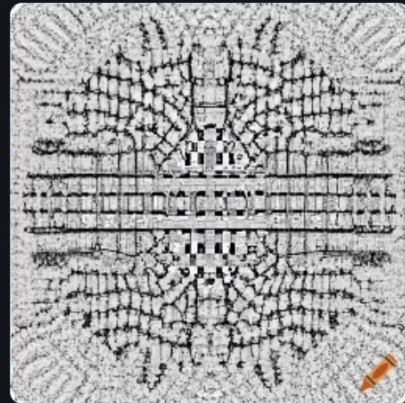
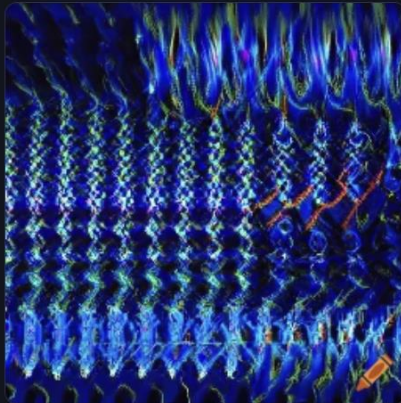
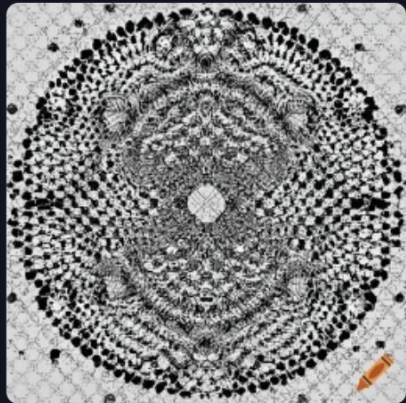
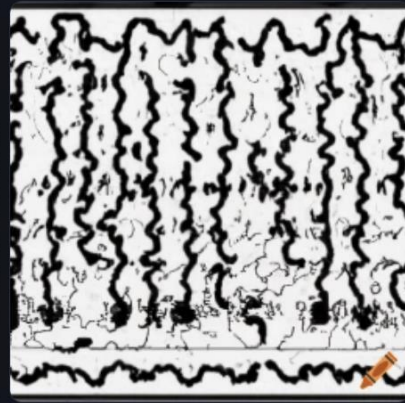
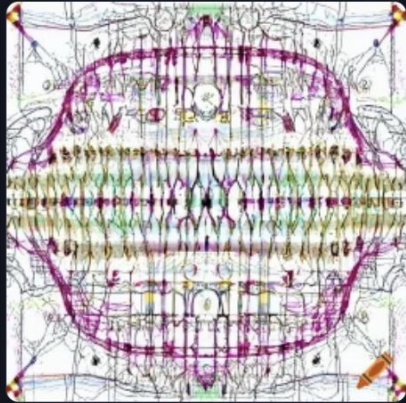
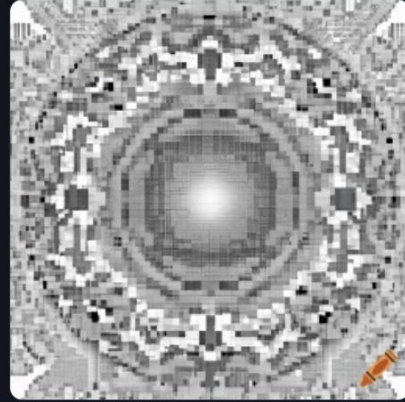
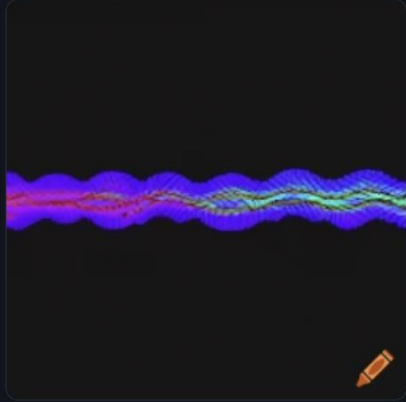


Draw

scatterplot of sine wave



Draw



Default (GPT-3.5)



draw a plot of a sine wave



AB

draw a plot of a sine wave



I'm a text-based AI and unable to create visual drawings or plots directly. However, I can describe how to create a simple sine wave plot using Python and the popular matplotlib library. You can run this code in a Python environment to generate a sine wave plot:



python

Copy code

```
import numpy as np
import matplotlib.pyplot as plt
```

```
# Create a time array
```

```
* np.pi, 1000) # Generate values from 0 to 2*pi
```

```
u can adjust the amplitude
```

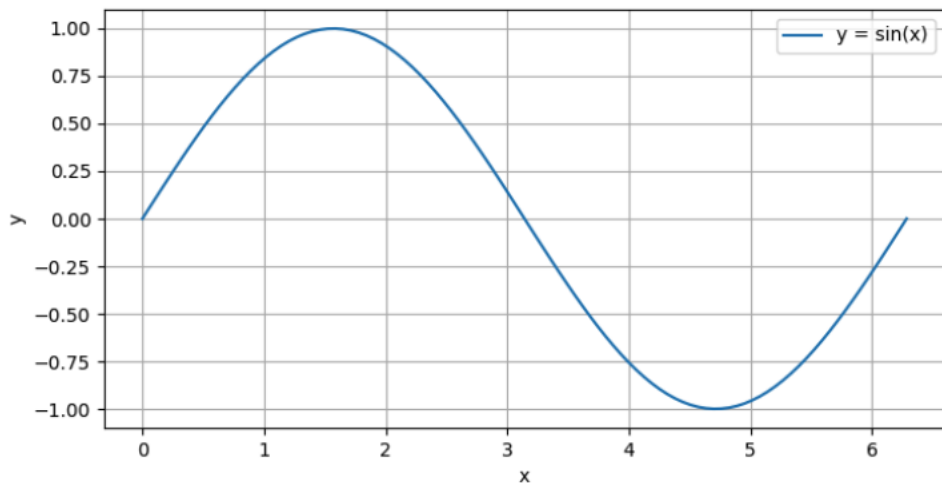
```
u can adjust the frequency
```

```
* np.sin(frequency * t)
```

```
4))
```

```
, label='Sine Wave')
```

Plot of a Sine Wave





draw a figure that could be at the top of a computer vision research paper



draw a figure that could be at the top of a computer vision research paper



Certainly! For a computer vision research paper, the top figure often highlights the core concept or architecture of the proposed method. Let's create a simple conceptual diagram of an object detection pipeline. This could show how an image is processed through a convolutional neural network (CNN) to produce bounding boxes and class labels.

Here's a basic example of such a diagram:

```
python Copy code

import matplotlib.pyplot as plt
from matplotlib.patches import Rectangle

# Create a new figure
fig, ax = plt.subplots(figsize=(10, 4))

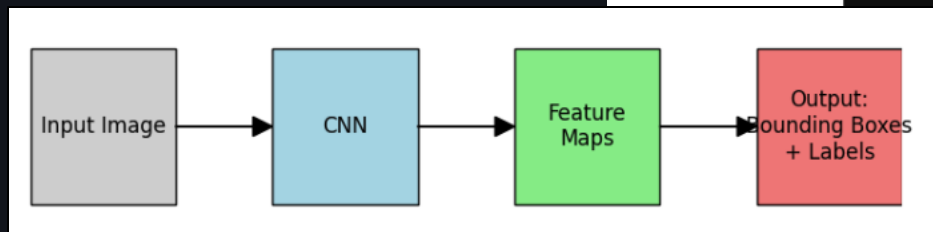
# Draw the image block
ax.add_patch(Rectangle((0.1, 0.3), 0.15, 0.4, edgecolor='black', facecolor='gray'))
plt.text(0.175, 0.5, 'Input Image', ha='center', va='center', fontweight='bold')

# Draw the CNN block
ax.add_patch(Rectangle((0.35, 0.3), 0.15, 0.4, edgecolor='black', facecolor='lightblue'))
plt.text(0.425, 0.5, 'CNN', ha='center', va='center', fontweight='bold', fontsize=12)
```

```
ax.add_patch(Rectangle((0.25, 0.5), 0.08, 0, head_width=0.05, head_length=0.02, facecolor='black'))

# Draw the Feature Maps block
ax.add_patch(Rectangle((0.5, 0.3), 0.15, 0.4, edgecolor='black', facecolor='lightgreen'))
plt.text(0.575, 0.5, 'Feature Maps', ha='center', va='center', fontweight='bold', fontsize=12)
```

ChatGPT



Instead of hoping to auto-generate our academic figures...  
... let's talk about the *process of designing* them

## Claim:

The *process of graphic design* is a **multi-objective** optimal control problem

*i.e. sequential decision-making problem!*

The *process of graphic design* is a **multi-objective** optimal control problem

## Thinking of it this way:

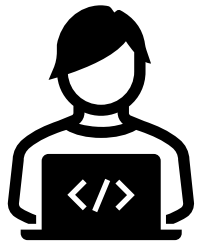
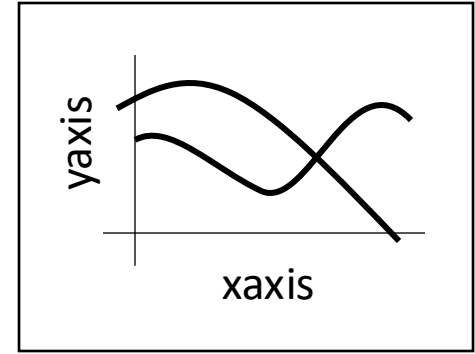
1. **Reduction to a known problem:** Makes the daunting challenge of design more “familiar”
2. **Reward design:** Lets us think about what **objectives** we seek to optimize with our designs
3. **Approximations:** Lets us derive practical design heuristics

# *Formalizing the Graphic Design Process*

*state*

$x$

*the design you see*



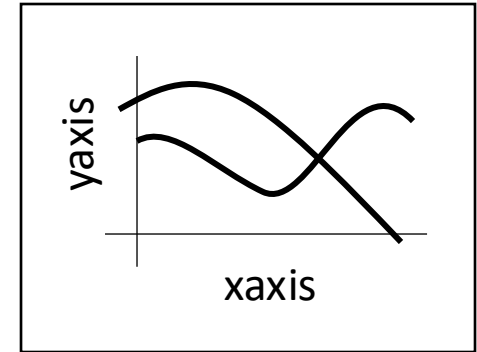
*You*  
*(the designer)*



*state*

$\mathcal{X}$

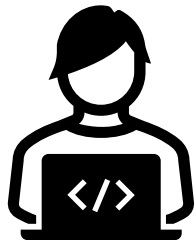
*the design you see*



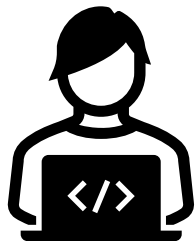
*control*

$\mathcal{U}$

$\in \{change_{font}, change_{color},$   
 $add_{line}, add_{text}, \dots\}$



*You*  
*(the designer)*

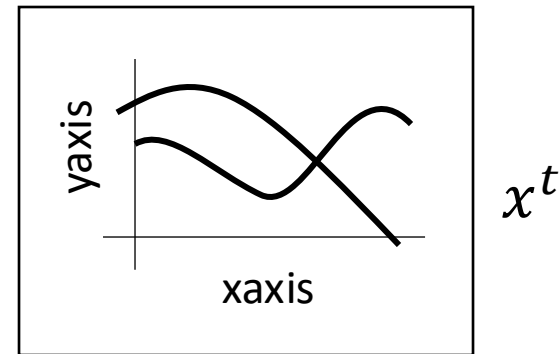


You  
(the designer)

state

$x$

the design you see



control

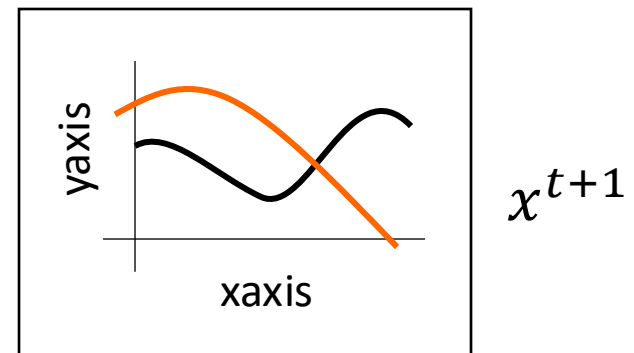
$u$

$\in \{change_{font}, change_{color},$   
 $add_{line}, add_{text}, \dots\}$

$u^t = change_{color}$

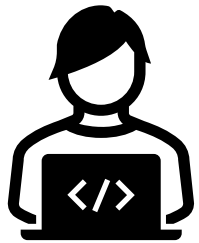
dynamics

$$x^{t+1} = f(x^t, u^t)$$



$$\max_{u^{0:T}}$$

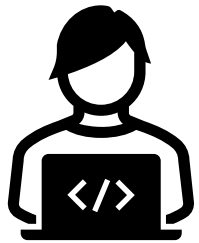
*sequence of design  
decisions*



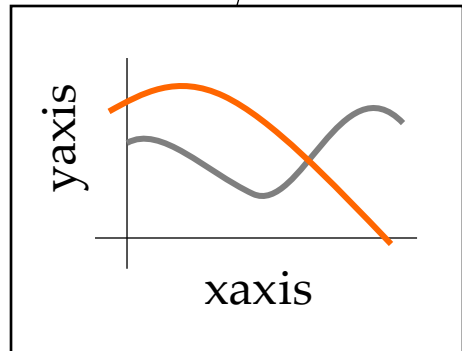
*You  
(the designer)*

*Information density  
of final design*

$$\max_{u^{0:T}} \text{Info}(x^T)$$



*You  
(the designer)*

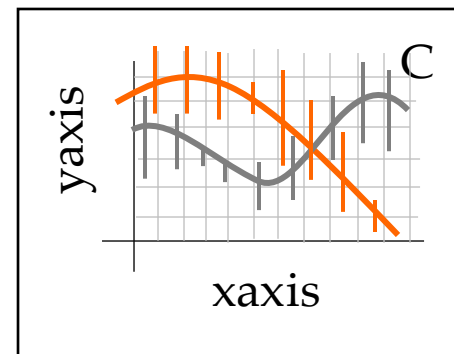
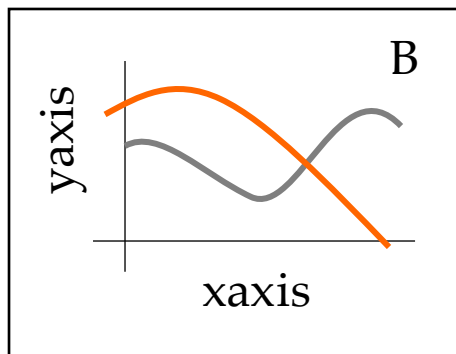
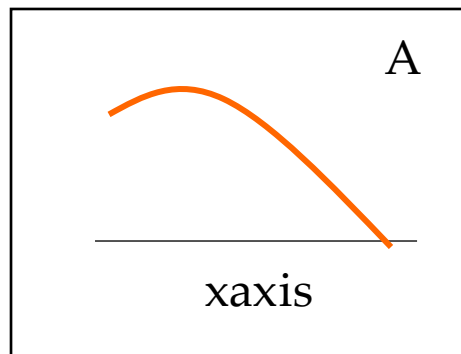


Information density  
of final design

$$\max_{u^{0:T}} \text{Info}(x^T)$$



Engineers typically don't  
have trouble with this 😊



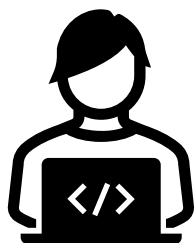
$$\text{Info}(x_A^T)$$

<

$$\text{Info}(x_B^T)$$

<

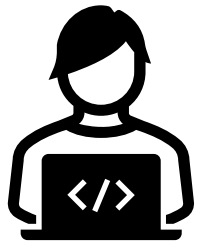
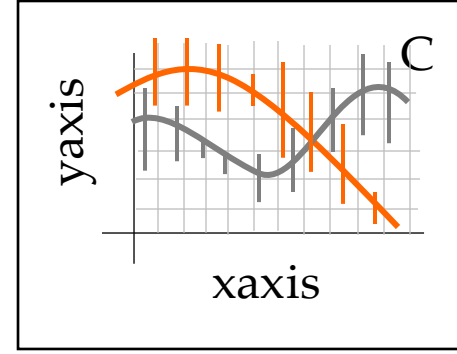
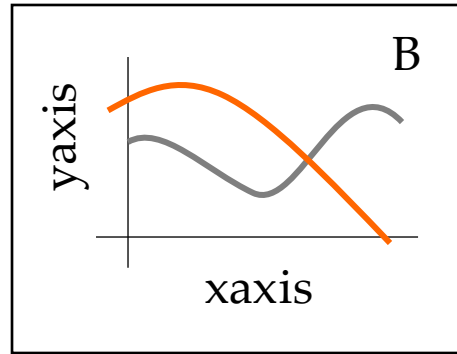
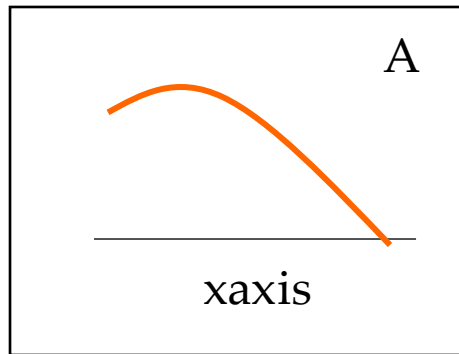
$$\text{Info}(x_C^T)$$



You  
(the designer)

*Viewer's cognitive load when parsing the final design*

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$



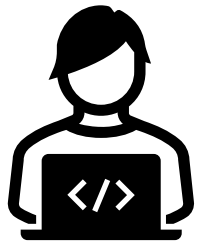
*You  
(the designer)*



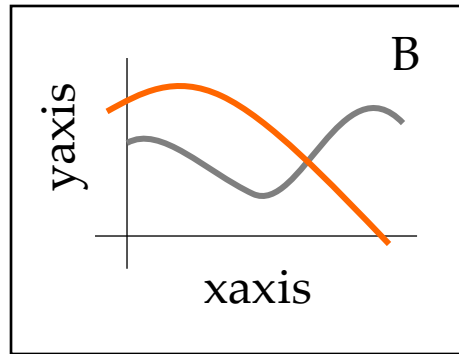
*The viewer*

*Viewer's cognitive load when parsing the final design*

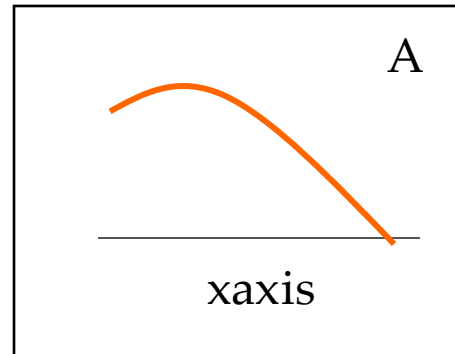
$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$



*You  
(the designer)*

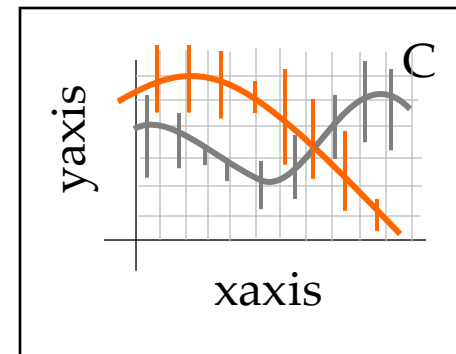


$\text{ViewerEffort}(x_B^T)$



$\text{ViewerEffort}(x_A^T)$

*Hard to tell what y-axis is*



$\text{ViewerEffort}(x_C^T)$

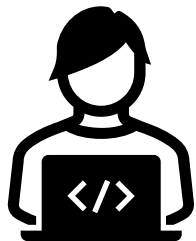
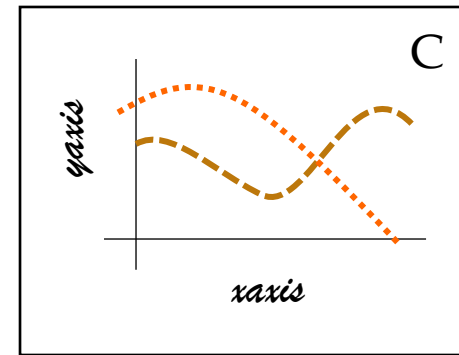
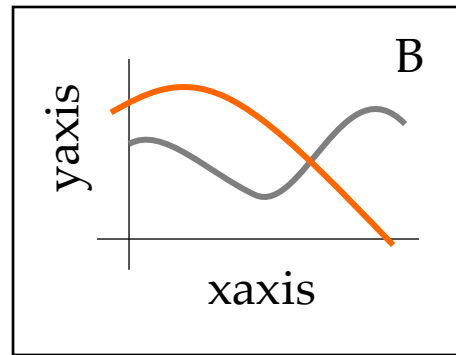
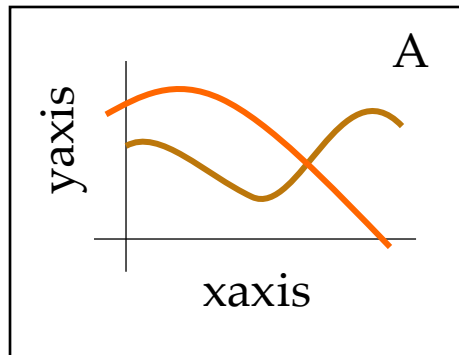
*Hard to focus on trends*



*The viewer*

*Viewer's cognitive load when parsing the final design*

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$



*You  
(the designer)*

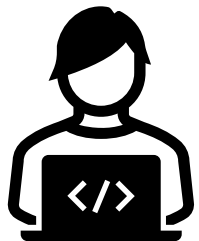


*The viewer*

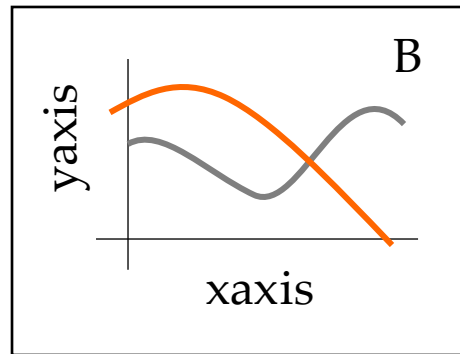


*Viewer's cognitive load when parsing the final design*

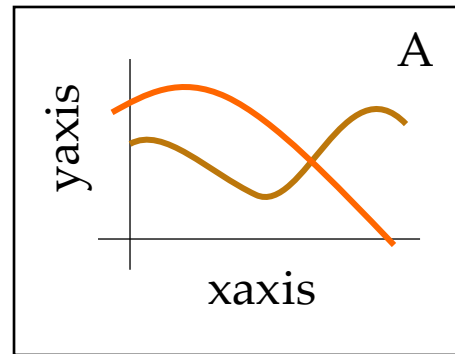
$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$



*You  
(the designer)*

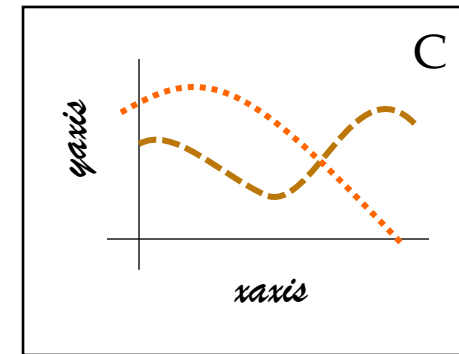


$\text{ViewerEffort}(x_B^T)$



$\text{ViewerEffort}(x_A^T)$

*Line colors blend together*



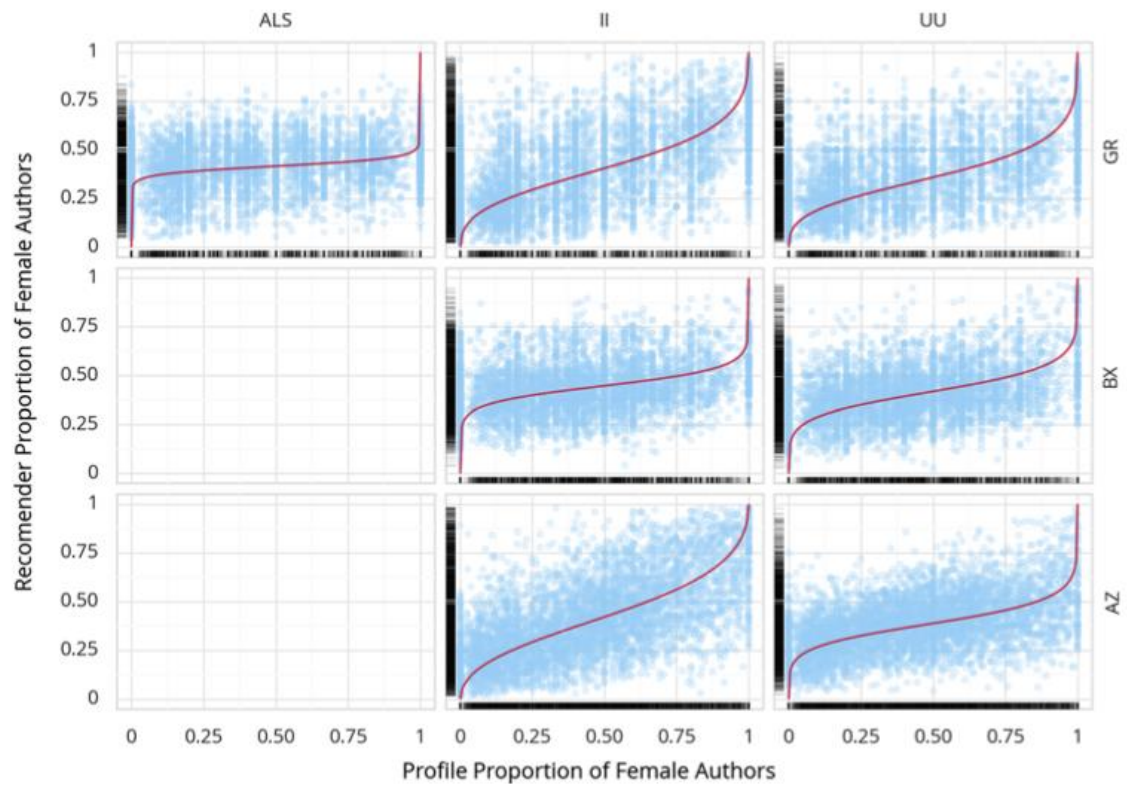
$\text{ViewerEffort}(x_C^T)$

*Hard to read font, line color and pattern*



*The viewer*

Figures from academic papers...



**Fig. 12** Scatter plots and regression curves for explicit feedback recommender response to individual users. Points are observed  $y/n$  proportions; curves are regression lines transformed from log-odds to proportions. Rug plots show marginal distributions

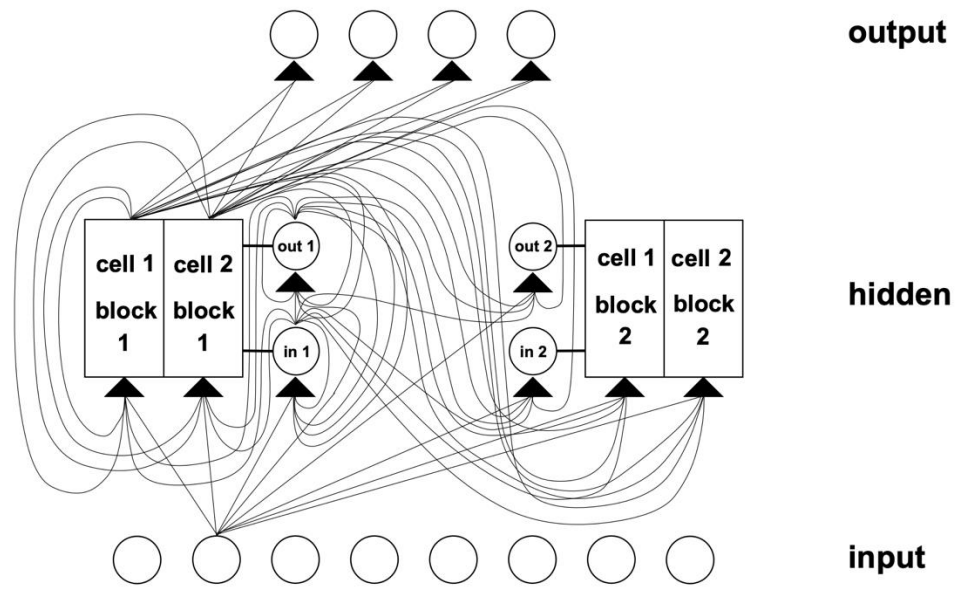





Figure 2: Example of a net with 8 input units, 4 output units, and 2 memory cell blocks of size 2. *in1* marks the input gate, *out1* marks the output gate, and *cell1/block1* marks the first memory cell of block 1. *cell1/block1*'s architecture is identical to the one in Figure 1, with gate units *in1* and *out1* (note that by rotating Figure 1 by 90 degrees anti-clockwise, it will match with the corresponding parts of Figure 1). The example assumes dense connectivity: each gate unit and each memory cell see all non-output units. For simplicity, however, outgoing weights of only one type of unit are shown for each layer. With the efficient, truncated update rule, error flows only through connections to output units, and through fixed self-connections within cell blocks (not shown here — see Figure 1). Error flow is truncated once it “wants” to leave memory cells or gate units. Therefore, no connection shown above serves to propagate error back to “” from which the connection originates (except for connections to output units), although



**Info( $x^T$ )** — **ViewerEffort( $x^T$ )**





The viewer

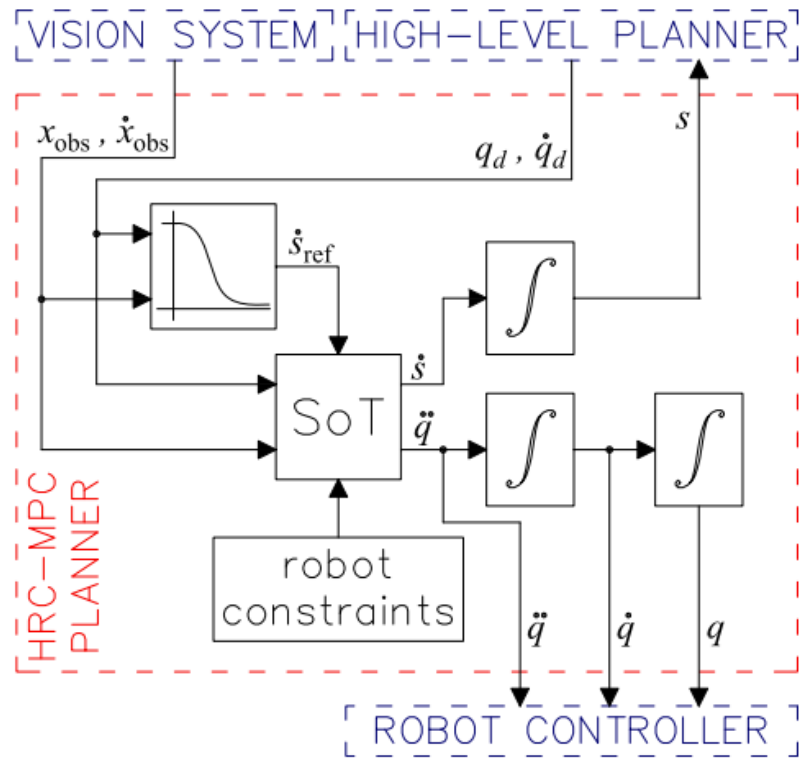
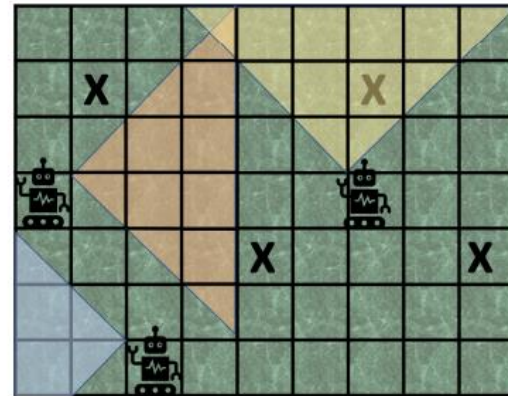
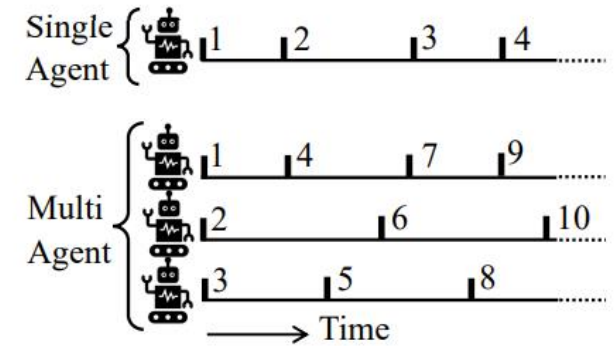


Fig. 1. Illustrative scheme of the proposed HRC-MPC planner.



(a) Multi-Agent Active search



(b) Single vs. Multi



The viewer

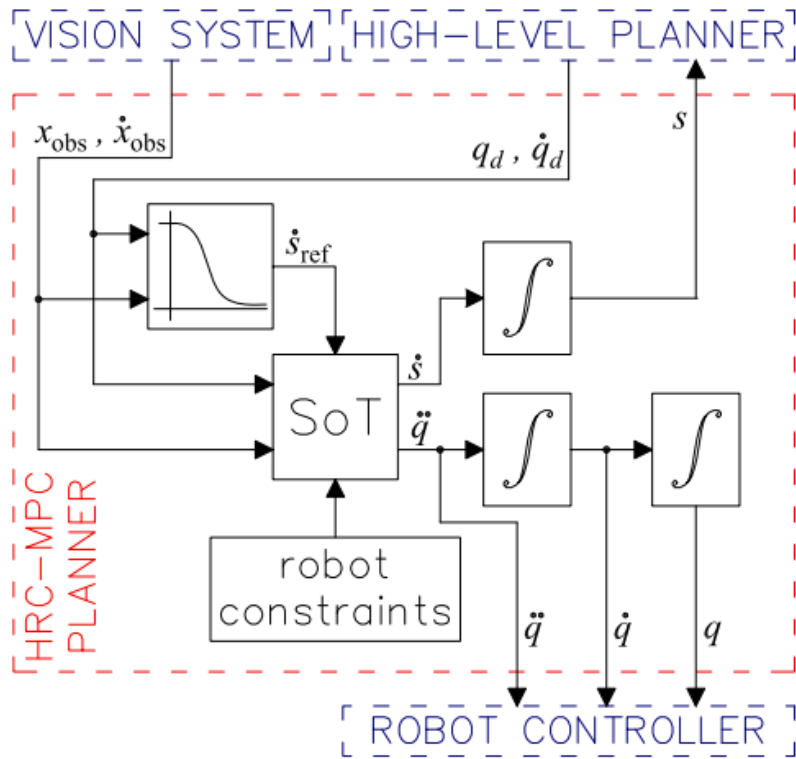
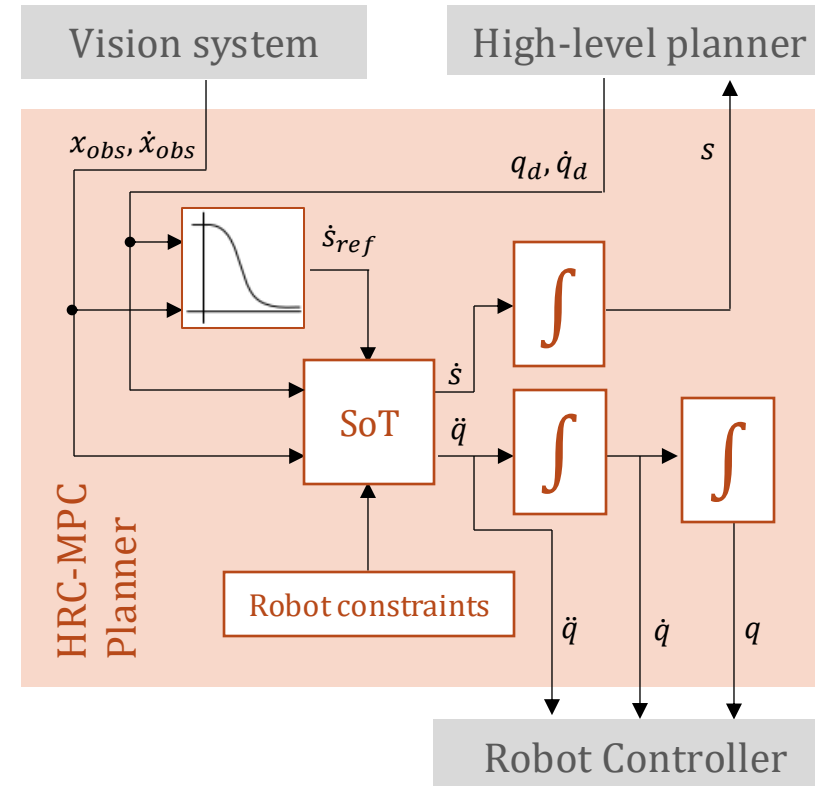


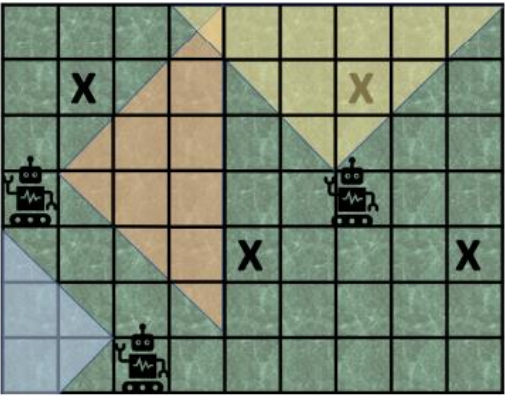
Fig. 1. Illustrative scheme of the proposed HRC-MPC planner.



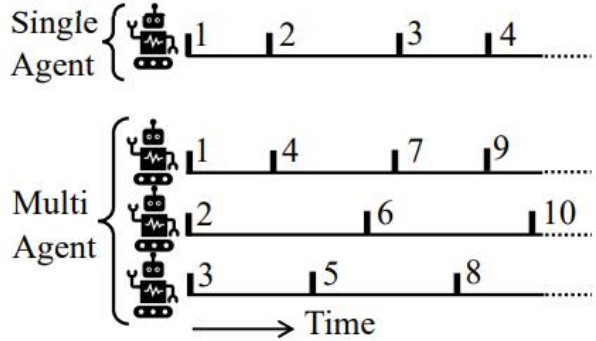
Easier to read font

Color highlights the main contribution

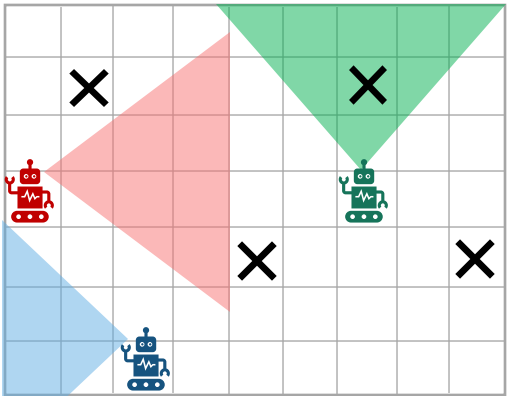
Color blocking shows relationships



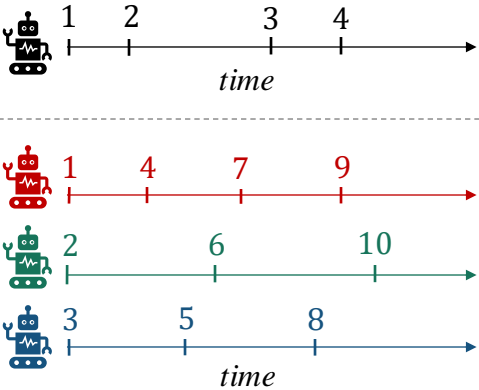
(a) Multi-Agent Active search



(b) Single vs. Multi



(a) Multi-Agent Active search



(b) Single vs. Multi

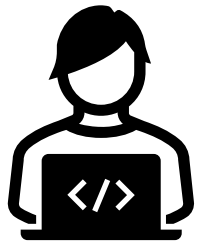
*No unnecessary background*

*Color has meaning (agent identity)*

*Remove unnecessary words*

*How do we design this reward function?*

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$



*You  
(the designer)*



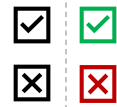
*The viewer*

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

low ViewerEffort



*contrasting colors*



*meaningful colors*



*reduce "visual clutter"*

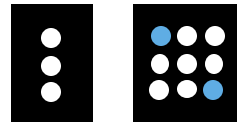


*strategic text, more icons*

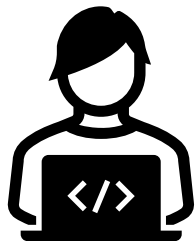
hello

*hello*

*readable fonts*



*alignment, balance-symmetry*



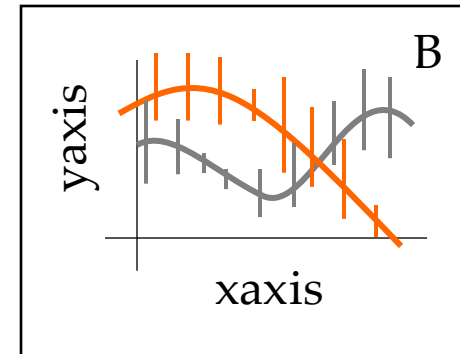
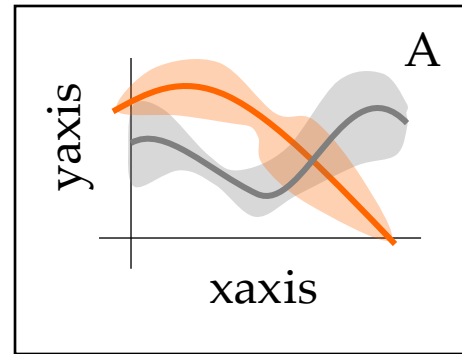
You  
*(the designer)*



*The viewer*

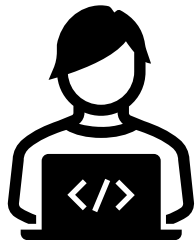
# Relationship between Info and ViewerEffort

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$



$$\text{Info}(x_A^T) = \text{Info}(x_B^T)$$

$$\text{ViewerEffort}(x_A^T) < \text{ViewerEffort}(x_B^T)$$



You  
(the designer)

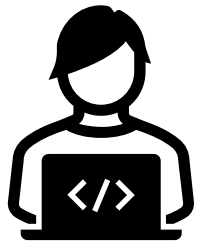
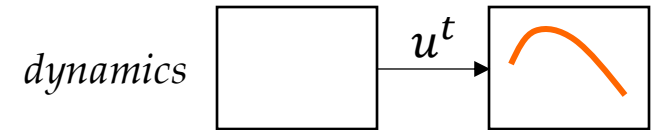


The viewer



$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

$$s. t. \quad x^{t+1} = f(x^t, u^t)$$



*You*  
*(the designer)*



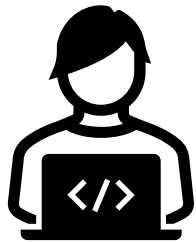
*The viewer*

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

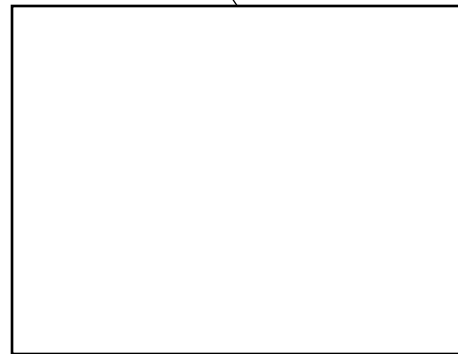
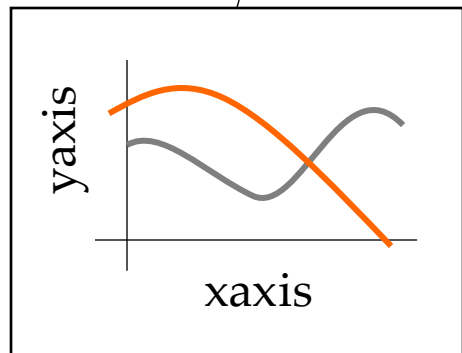
s. t.

$$x^{t+1} = f(x^t, u^t)$$

$$x^0 = x_{\text{blank}}$$



You  
(the designer)



The viewer

*This is a sequential decision-making problem  
(i.e., optimal control problem)!*



$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

$$\begin{aligned} \text{s. t.} \quad & x^{t+1} = f(x^t, u^t) \\ & x^0 = x_{\text{blank}} \end{aligned}$$

*But, challenging because design space is huge...*

# *Tractable Approximations*

# *Dimensionality Reduction*

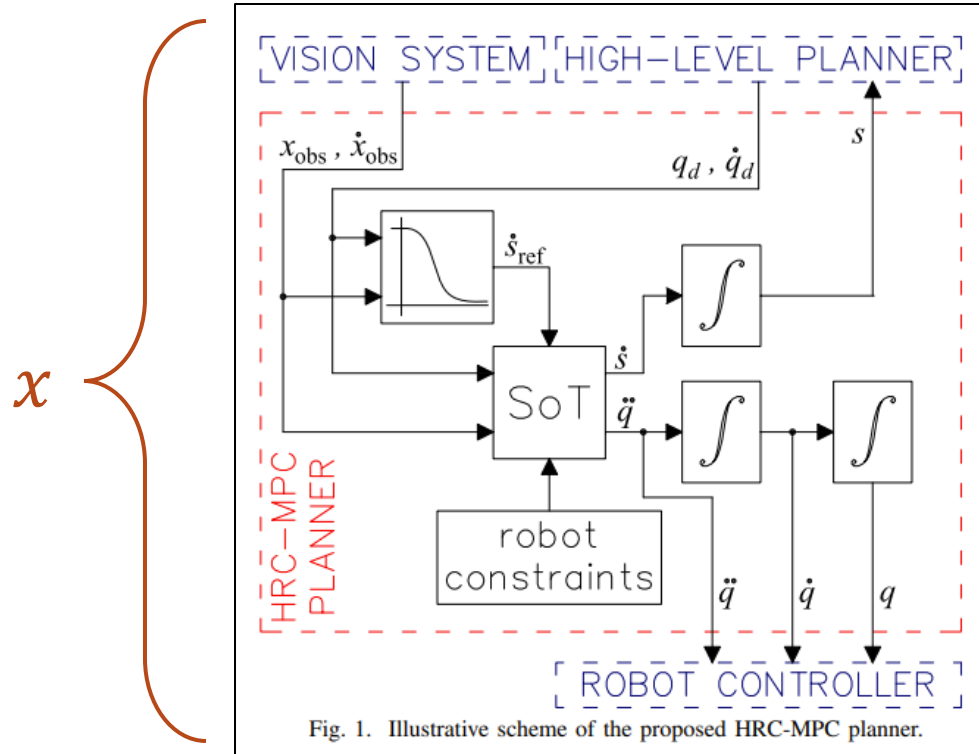
*Reduce the size of the design “state” space*

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

$$s. t. \quad \begin{aligned} x^{t+1} &= f(x^t, u^t) \\ x^0 &= x_{blank} \end{aligned}$$

# Dimensionality Reduction

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$



# Dimensionality Reduction

$$\max_{u^{0:T}} \text{Info}(\underbrace{\mathcal{E}(x^T)}_{\text{encoded design}}) - \text{ViewerEffort}(\underbrace{\mathcal{E}(x^T)}_{\text{encoded design}})$$

$$\mathcal{E}: \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ where } n \gg m$$

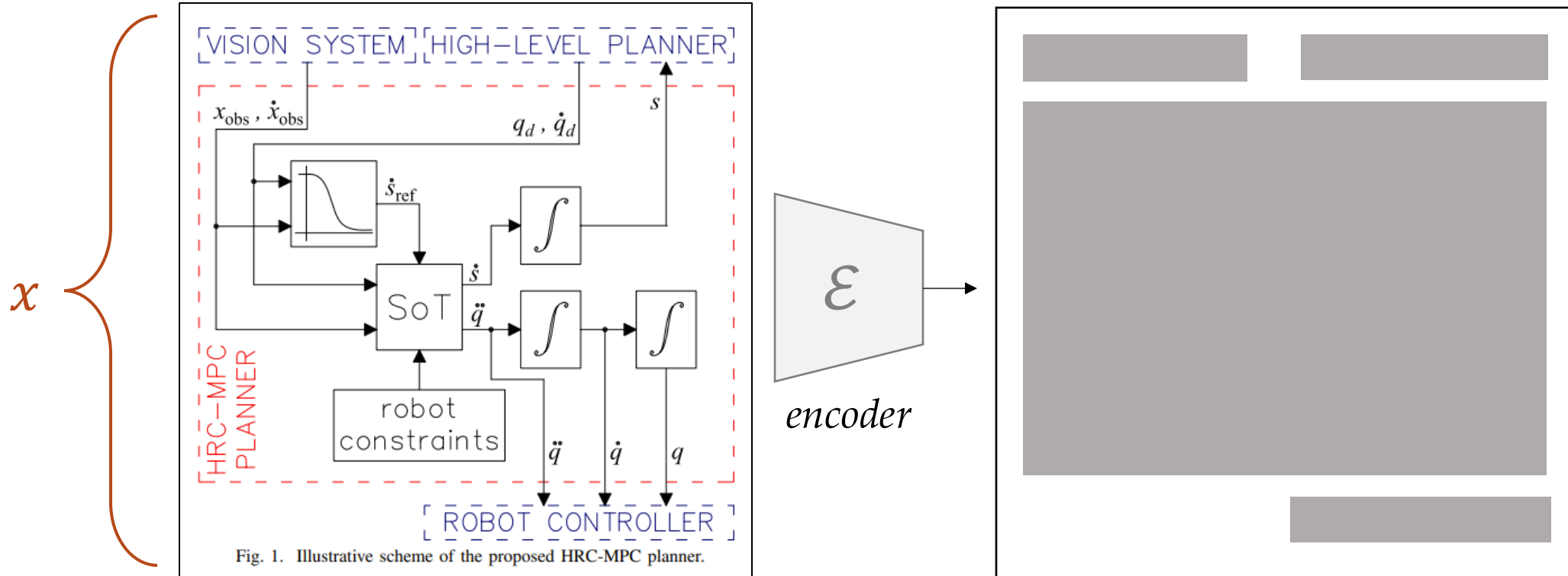
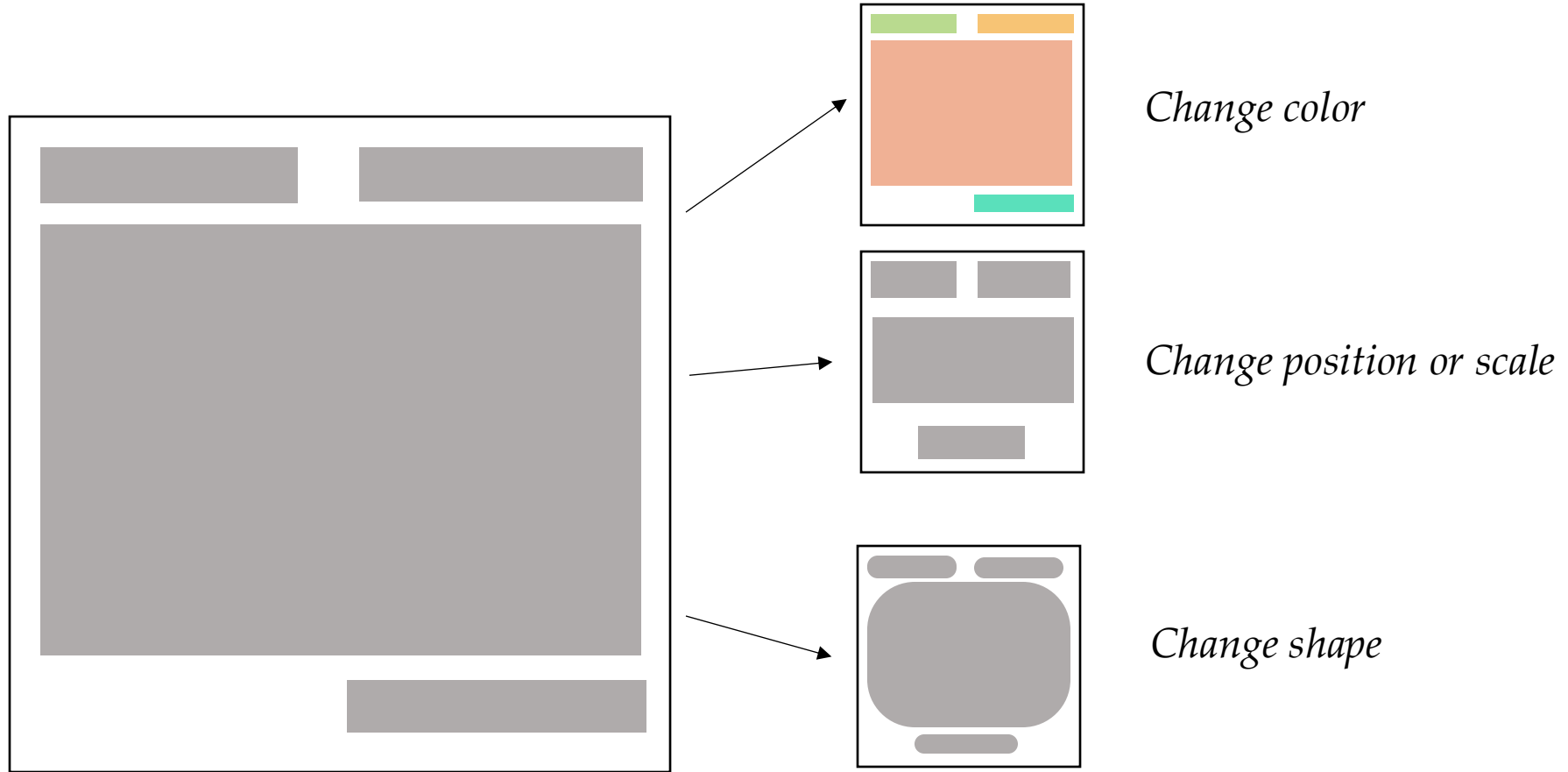


Fig. 1. Illustrative scheme of the proposed HRC-MPC planner.

Consider only the “primitives” of the design

# Dimensionality Reduction

$$\max_{u^{0:T}} \text{Info}(\mathcal{E}(x^T)) - \text{ViewerEffort}(\mathcal{E}(x^T))$$



*It's easier to reason about how "actions" influence the design*



# Dimensionality Reduction

$$\max_{u^{0:T}} \text{Info}(\mathcal{E}(x^T)) - \text{ViewerEffort}(\mathcal{E}(x^T))$$

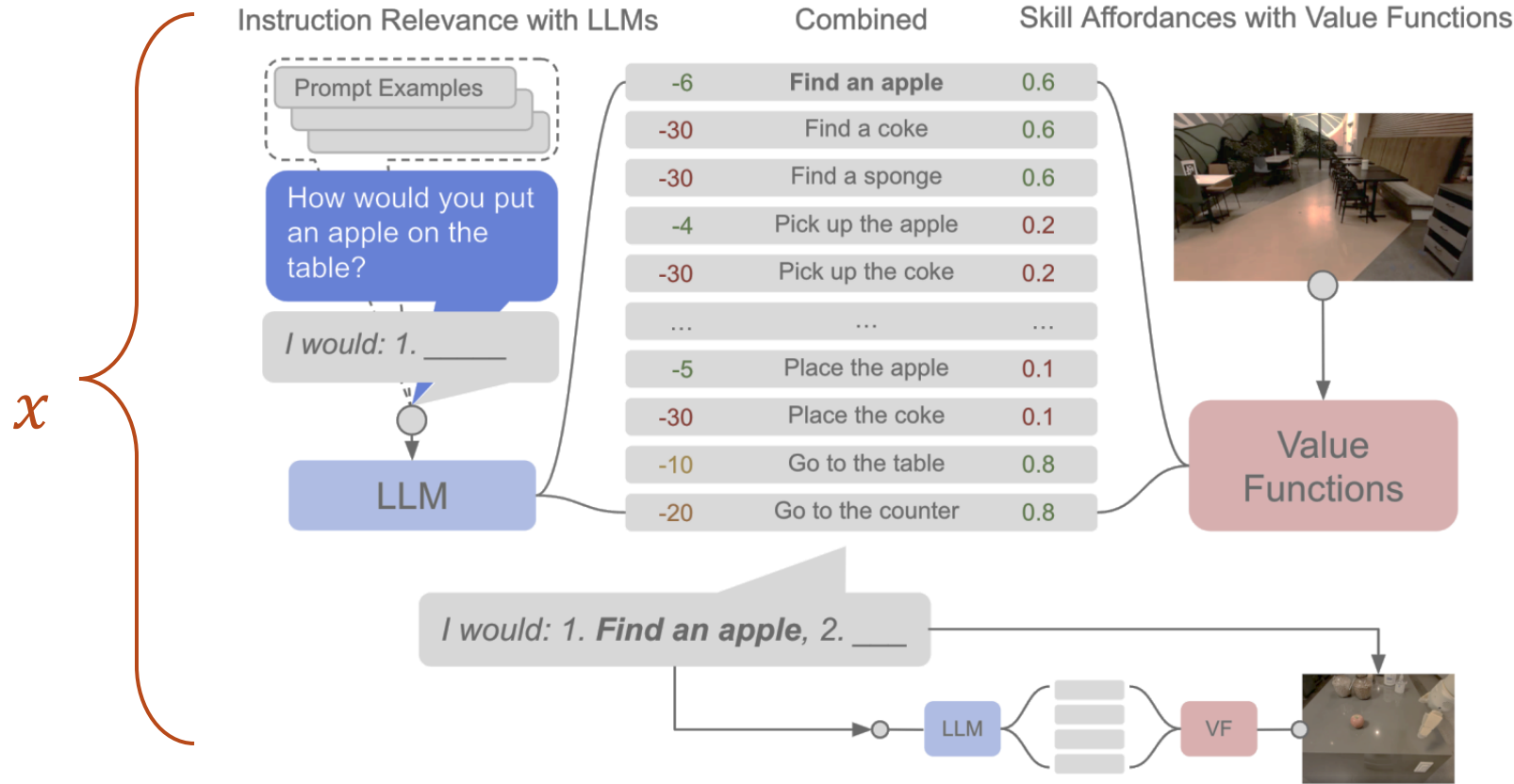
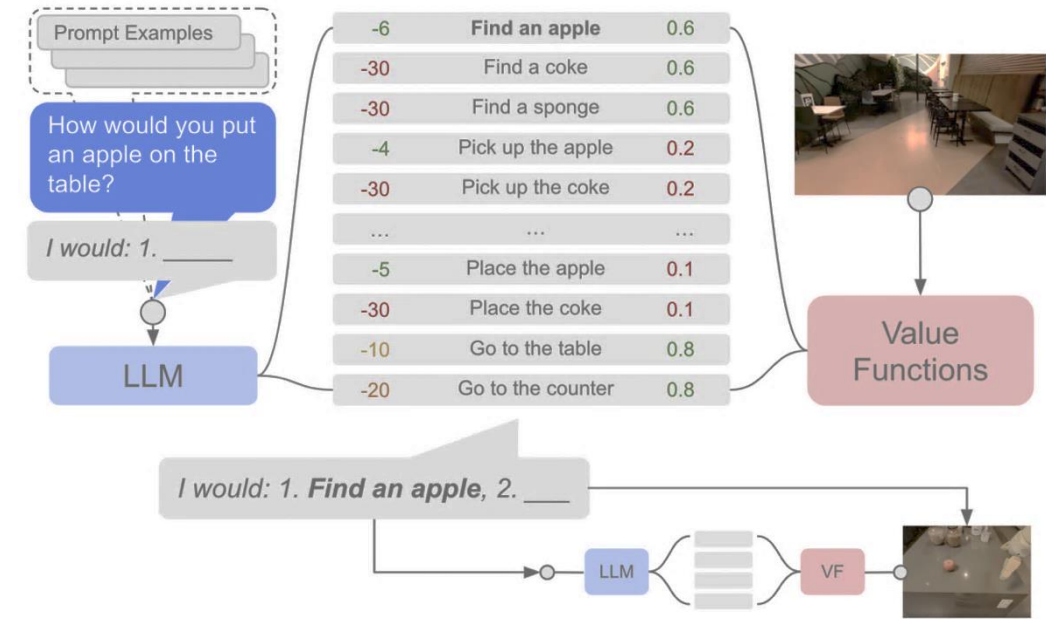


Figure 3: Given a high-level instruction, SayCan combines probabilities from a LLM (the probability that a skill is useful for the instruction) with the probabilities from a value function (the probability of successfully executing said skill) to select the skill to perform. This emits a skill that is both possible and useful. The process is repeated by appending the skill to the response and querying the models again, until the output step is to terminate. Appendix Figures 12 and 2 focus on the LLM and VFS components.

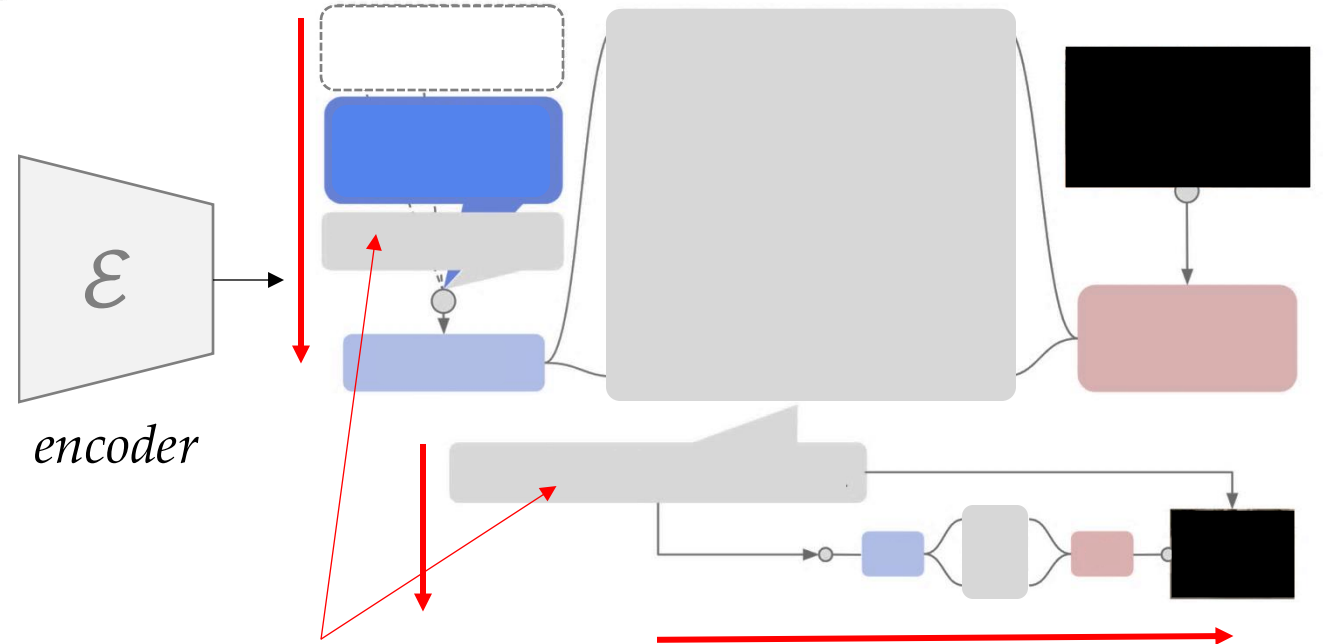
# Dimensionality Reduction

$$\max_{u^{0:T}} \text{Info}(\mathcal{E}(x^T)) - \text{ViewerEffort}(\mathcal{E}(x^T))$$

Instruction Relevance with LLMs      Combined      Skill Affordances with Value Functions



*Flow problem: need to read top-down and left-right*

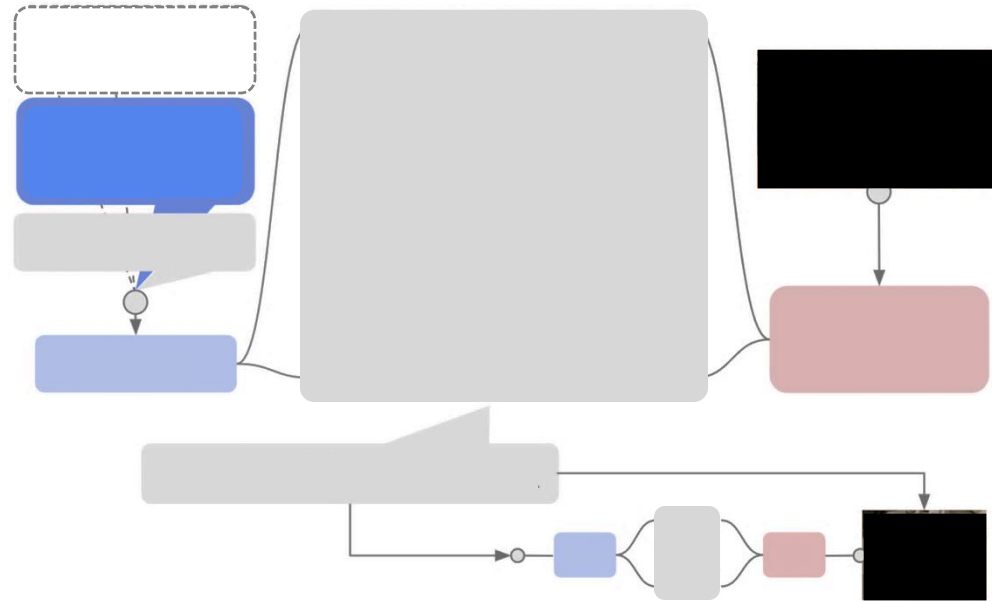


*Missing connection*

*Visual repetition*

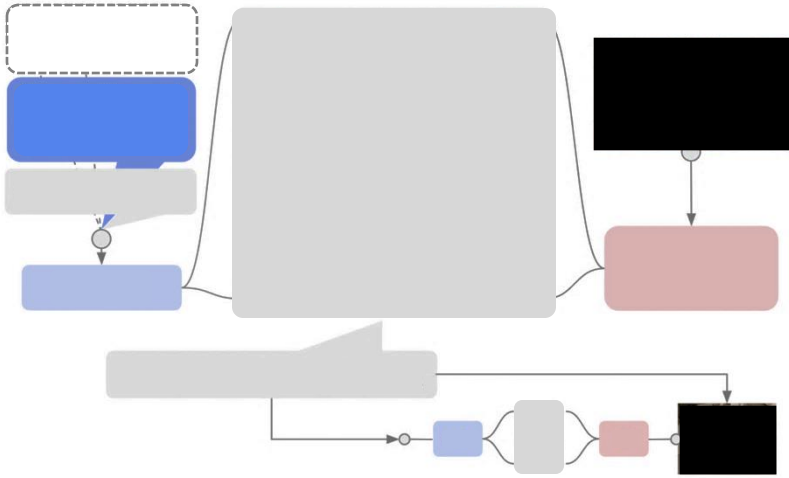
# *Dimensionality Reduction*

$$\max_{u^{0:T}} \text{Info}(\mathcal{E}(x^T)) - \text{ViewerEffort}(\mathcal{E}(x^T))$$

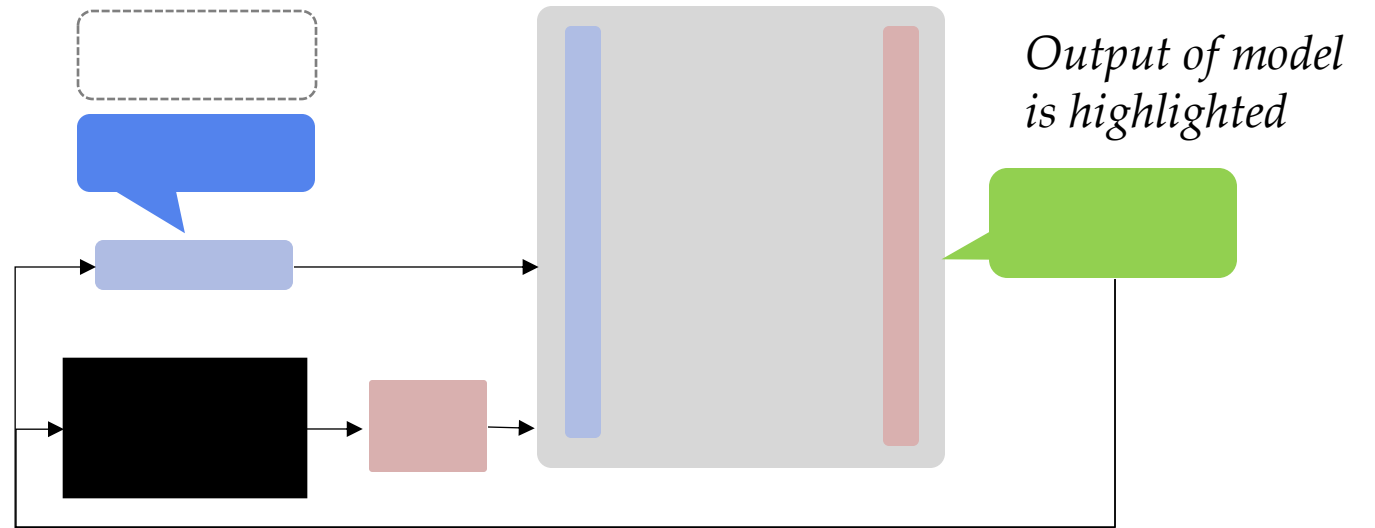


# Dimensionality Reduction

$$\max_{u^{0:T}} \text{Info}(\mathcal{E}(x^T)) - \text{ViewerEffort}(\mathcal{E}(x^T))$$



*Use color to connect inner and outer components*



*Output of model is highlighted*

*Use "feedback" arrows to show the repetition instead of repeating the graphic*

# *Minimize Control Effort*

*...by sketching before generating final figures*

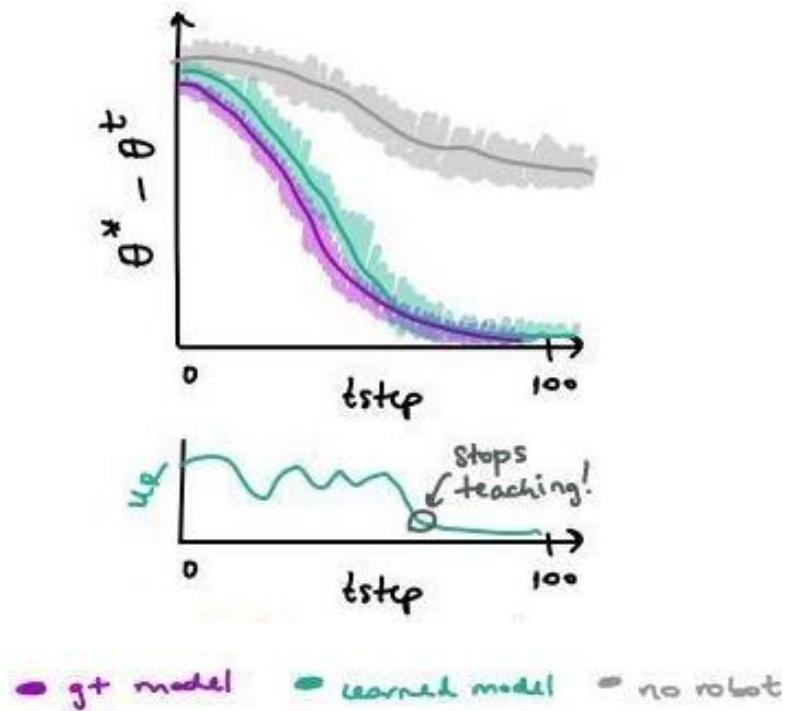
$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

$$\begin{aligned} s. t. \quad & x^{t+1} = f(x^t, u^t) \\ & x^0 = x_{blank} \end{aligned}$$

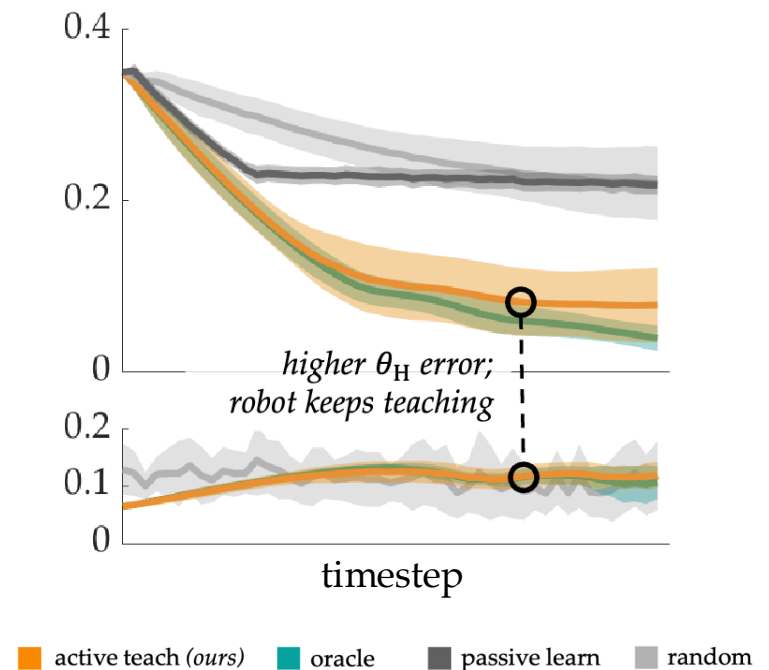
# Minimize Control Effort

...by sketching before programmatically generating figures

## Sketch



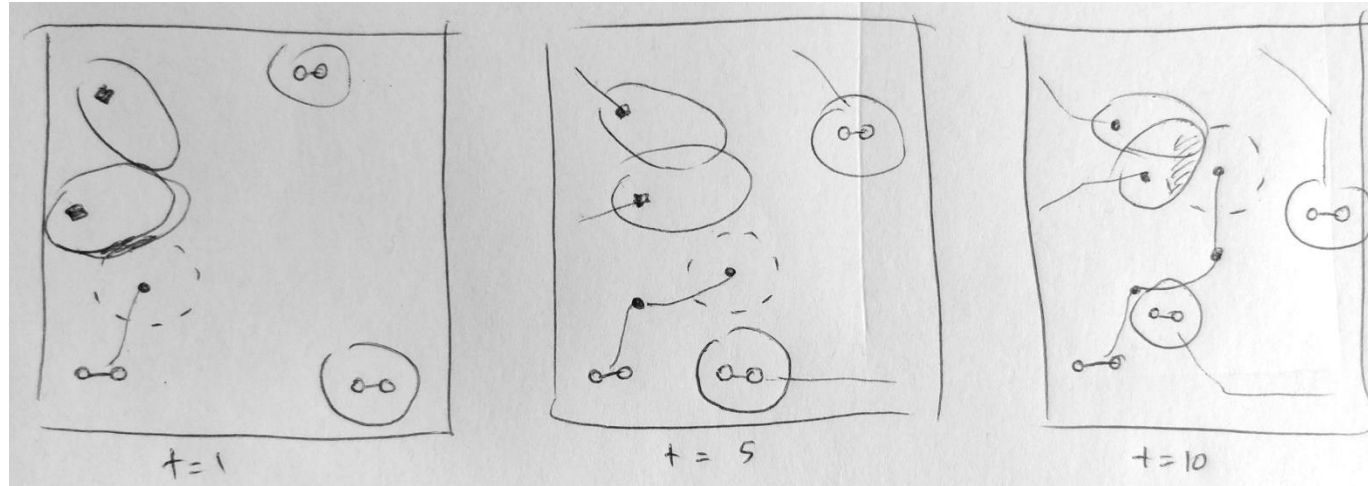
## Final Design



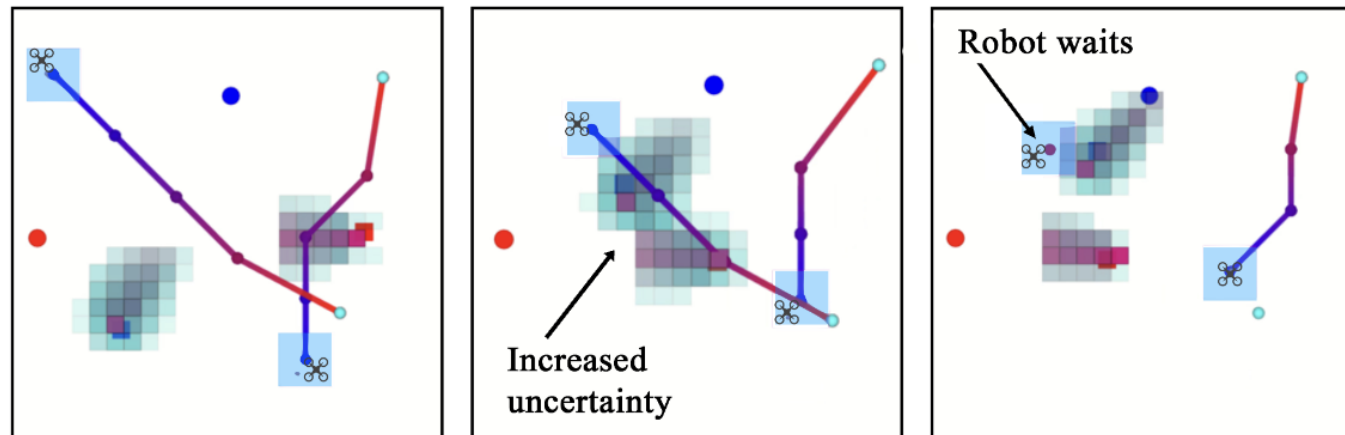
# Minimize Control Effort

*...by sketching before running expensive experiments*

Sketch



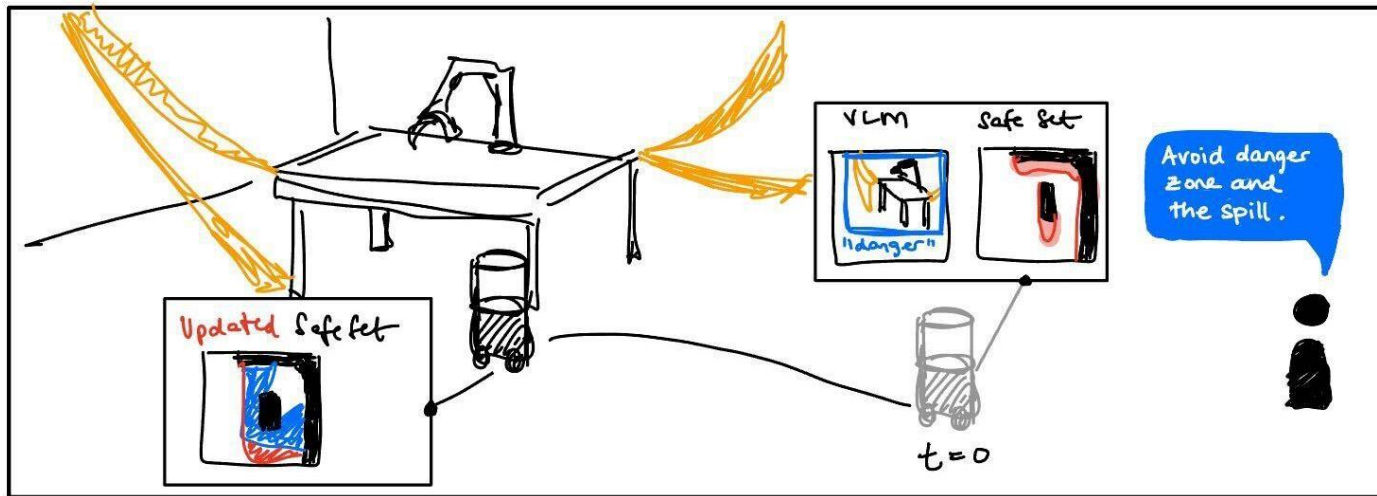
Final Design



# Minimize Control Effort

...by sketching before running expensive experiments

Sketch



Final Design

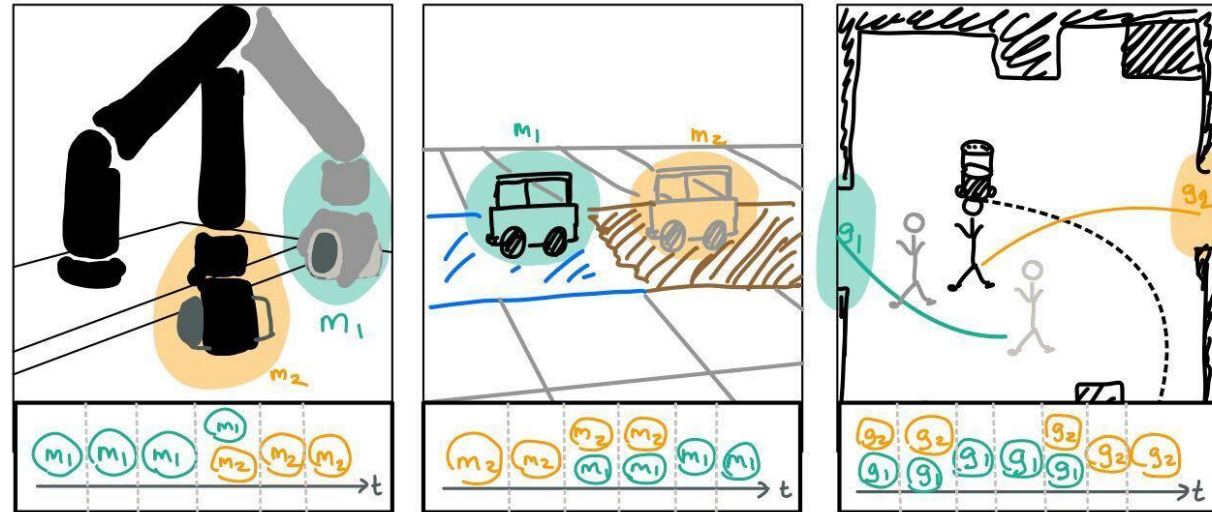




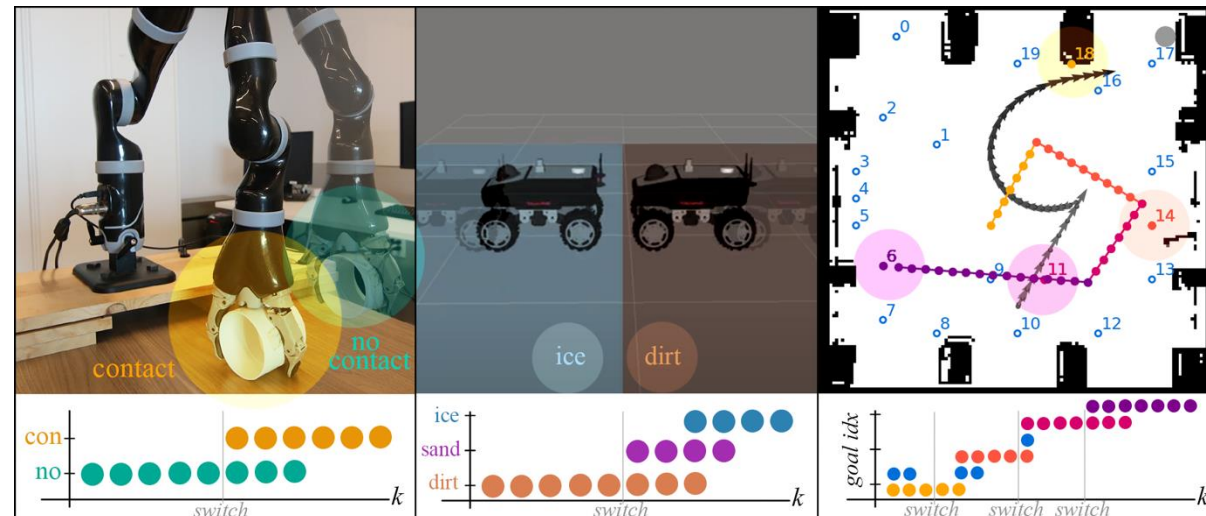
# Minimize Control Effort

...by sketching before running final simulations

Sketch



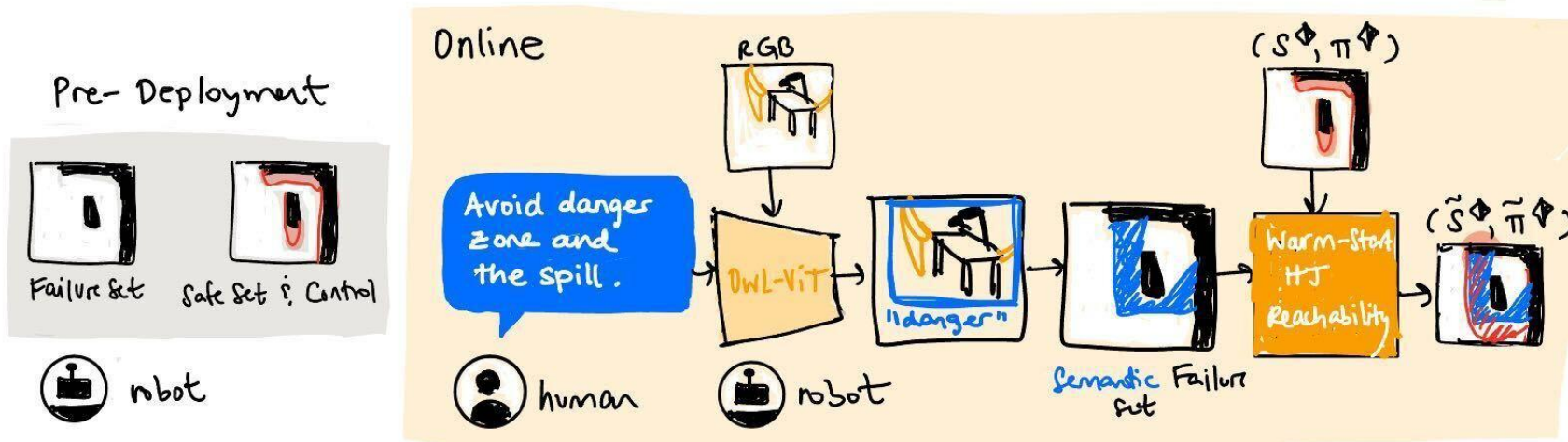
Final Design



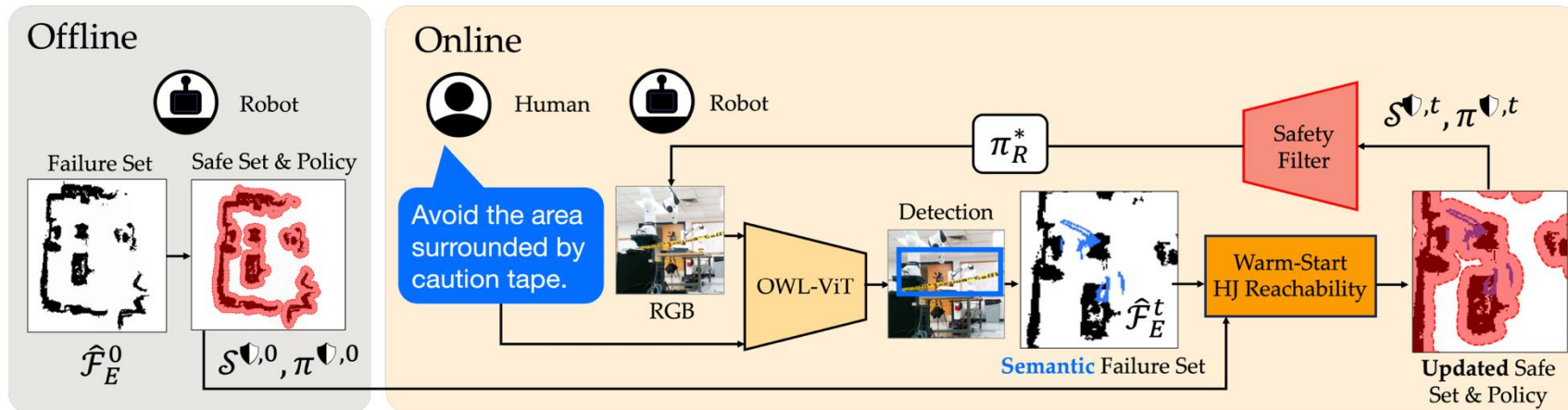
# Minimize Control Effort

...by sketching before making final figures in PPT

Sketch



Final Design



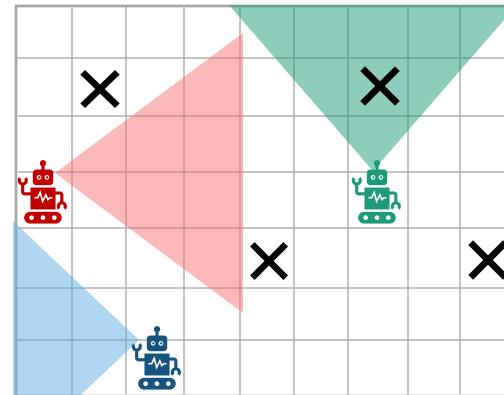
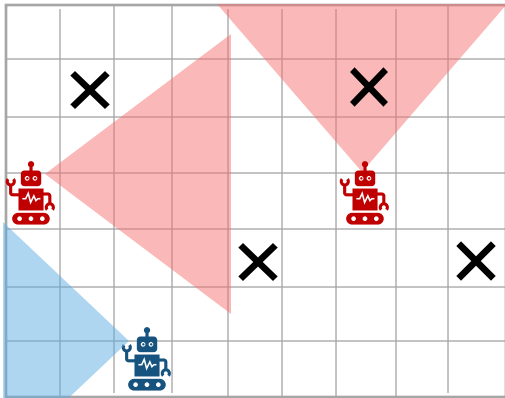
# *Greedy Approximation*

*Plan for greedily for one-step, and replan*

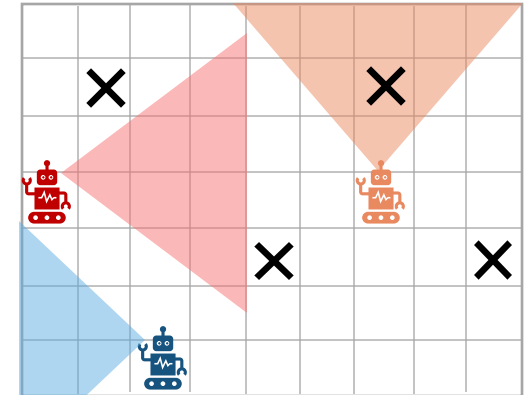
$$\max_{u^0} \text{Info}(x^1) - \text{ViewerEffort}(x^1)$$

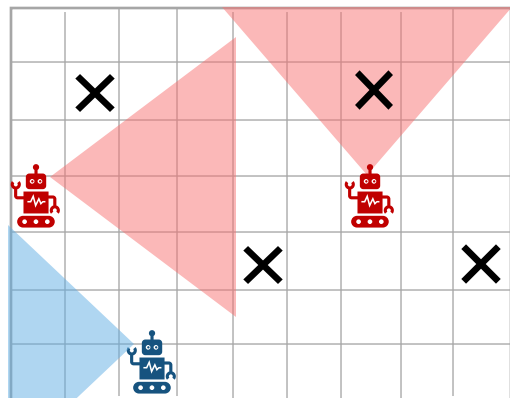
$$\begin{aligned} s. t. \quad & x^1 = f(x^0, u^0) \\ & x^0 = x_{init} \end{aligned}$$

*When designing, “simulate” an action and compare*



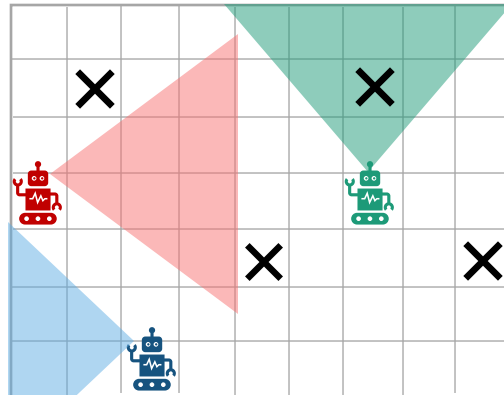
*vs.*





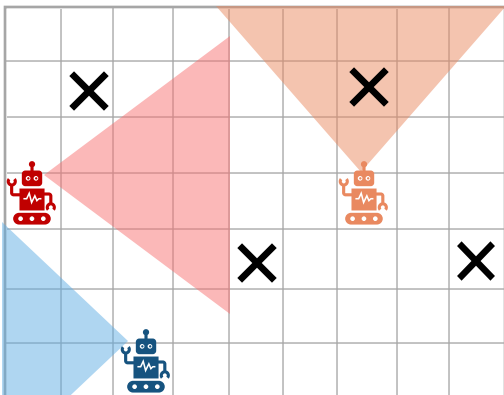
$x^0$

$u = color_{gr}$



$x^1 = f(x^0, color_{gr})$

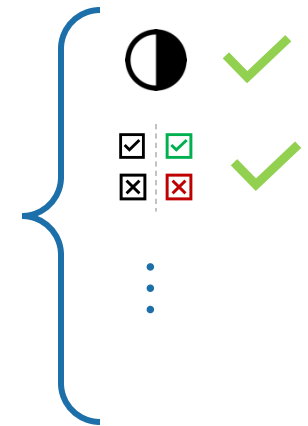
$u = color_{or}$



$x^1 = f(x^0, color_{or})$

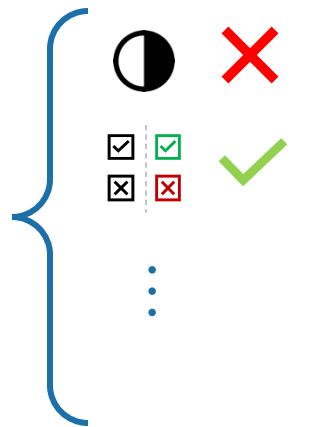
$Info(x^1) = 5$

$ViewEff(x^1) = 3$



$Info(x^1) = 5$

$ViewerEffort(x^1) = 6$



# *Visualizing Space & Time*



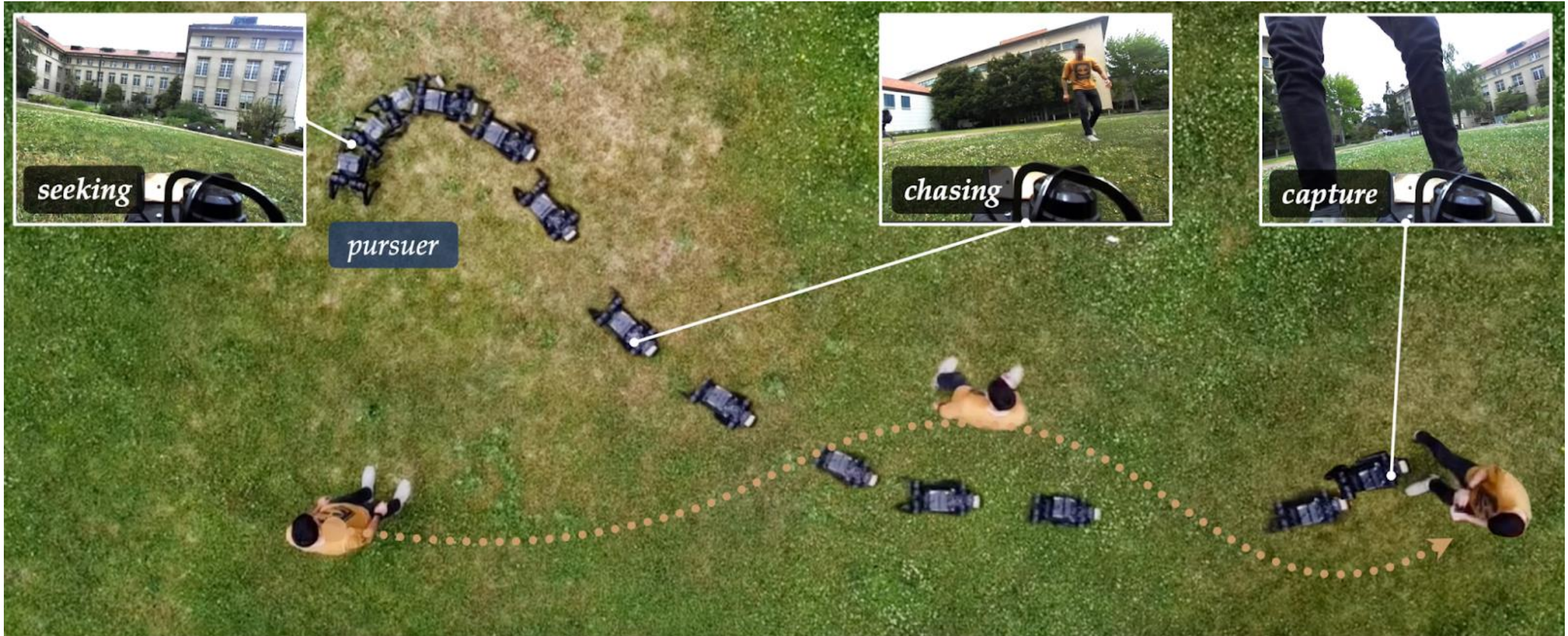
# *Visualizing Space & Time*

*...by using screenshot overlays*



# Visualizing Space & Time

*...by using screenshot overlays*

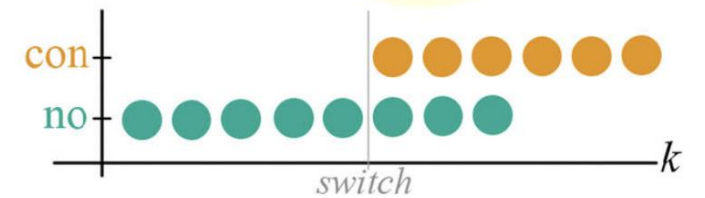
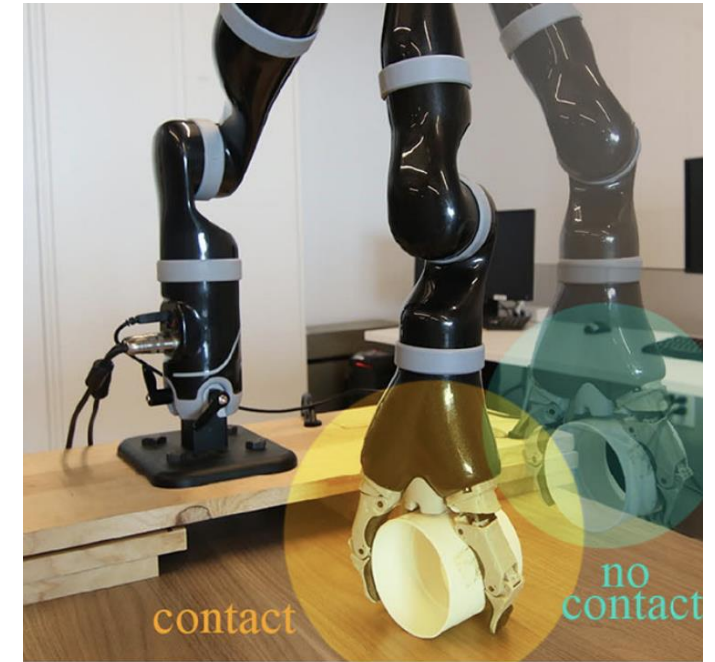
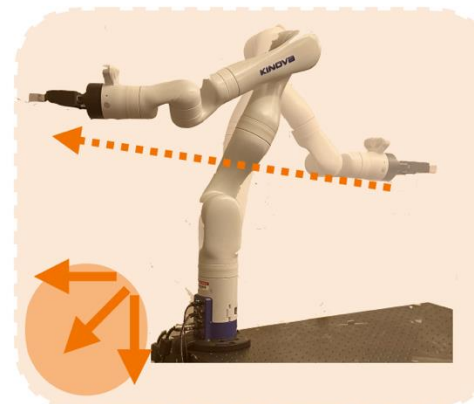
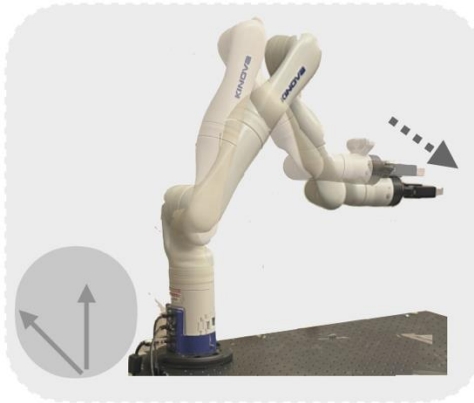
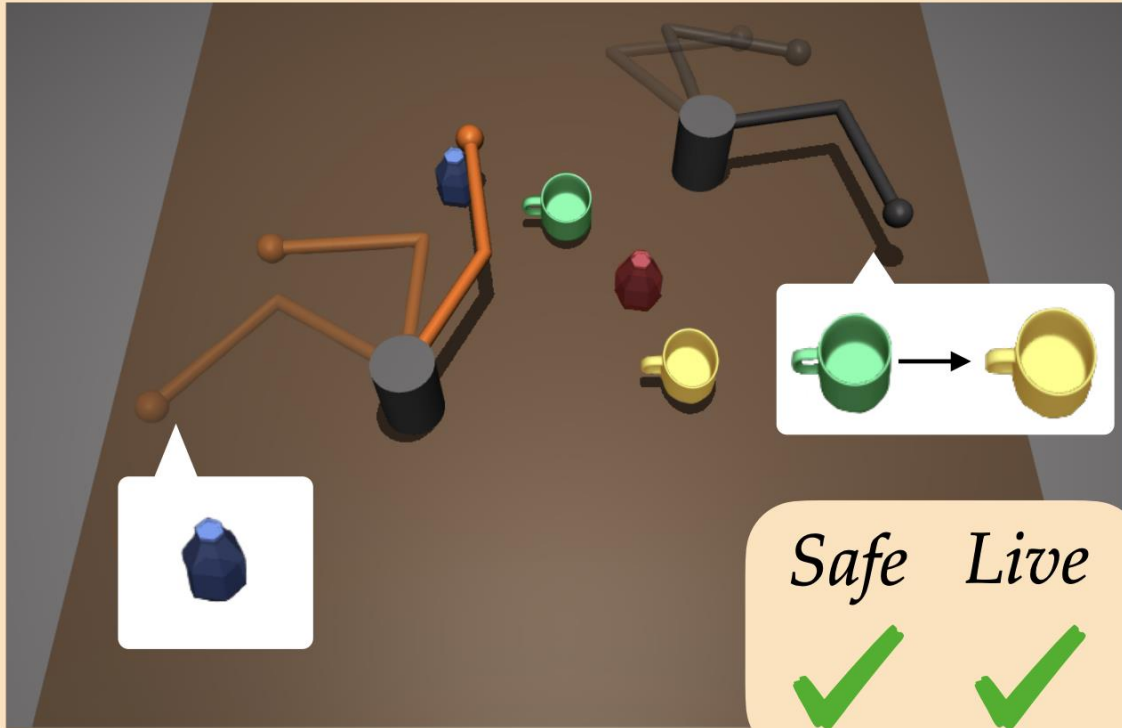




# Visualizing Space & Time

...by using transparency

SLIDE



# *Visualizing Space & Time*

*...by using colored trajectory overlays*



# Visualizing "human-centric" Data

... by using common iconography

## Language Feedback

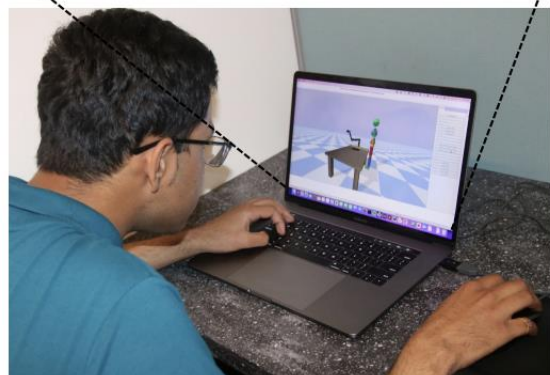
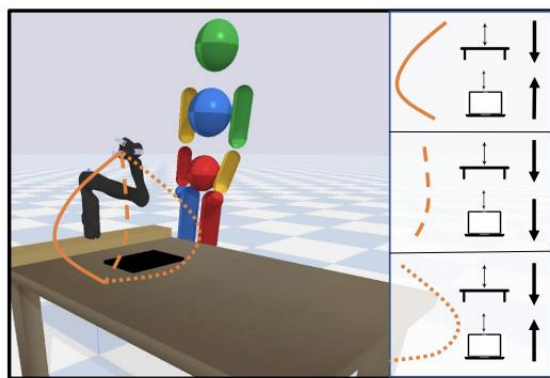
Sent at 12:04

Avoid the area  
surrounded by  
caution tape



Sent at 12:05

Avoid the coffee  
spill



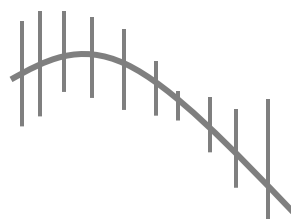
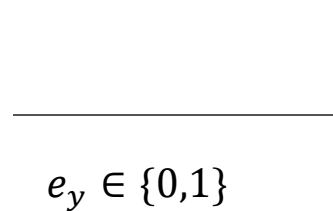
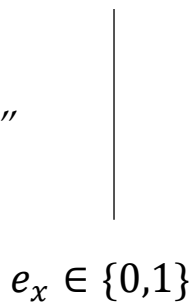
Bobu, A., et al. "Sirl: similarity-based implicit representation learning." HRI 2023.



Mullen, James F., et al. "Communicating inferred goals with passive augmented reality and active haptic feedback." RA-L. (2021)

# *A Simple Example & Numerical Solution*

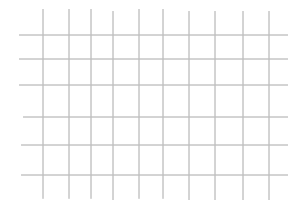
"primitives"



$e_A \in \{0,1\}$   
 $c_A \in \{g, o\}$   
 $p_A \in \{-, - -\}$   
 $s_A \in \{0,1\}$



$e_B \in \{0,1\}$   
 $c_B \in \{g, o\}$   
 $p_B \in \{-, - -\}$   
 $s_B \in \{0,1\}$



$e_{grid} \in \{0,1\}$

state  $x = [e_x, e_y, e_A, c_A, p_A, s_A, e_B, c_B, p_B, s_B, e_{grid}]$

control  $u \in \{[1,0,0,0,0,0,0,0,0,0,0], \dots [0,0,0,0,0,0,0,0,0,0,1]\}$

dynamics  $f(x, u) = (x + u) \% 2$

objective

$\text{Info}(x) = \|x\|_2$

$\text{ViewerEffort}(x) = \begin{cases} +10 \\ -10 \\ -\|c_A - c_B\|_2 \\ \|p_A - p_B\|_2 \\ \|s_A - s_B\|_2 \\ \dots \end{cases}$

*if I don't have axes or data, its more effortful to interpret results*

*if  $e_i = 0, i \in \{x, y, A, B\}$*

*if  $e_i = 1$*

*more different colors, less effort*

*more similar the pattern, less effort*

*more similar std. dev., less effort*

$\max_{u^0} \text{Info}(x^1) - \text{ViewerEffort}(x^1)$

```
optimal_designer.ipynb ☆
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

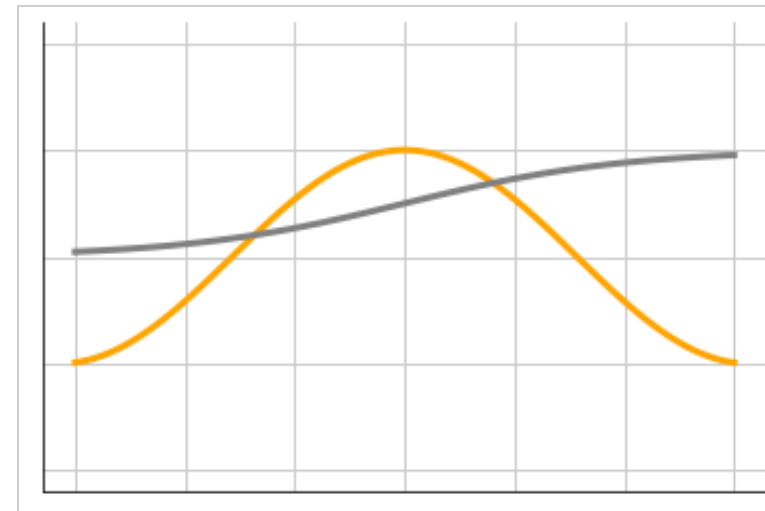
[87] 42 return r_info(x) - r_viewer_effort(x)

[88] 1 def get_actions(x):
2     action_list = []
3     action_list.append(np.zeros(len(x))) # add zero control
4     for i in range(len(x)):
5         u = np.zeros(len(x))
6         u[i] = 1
7         action_list.append(u)
8     return action_list

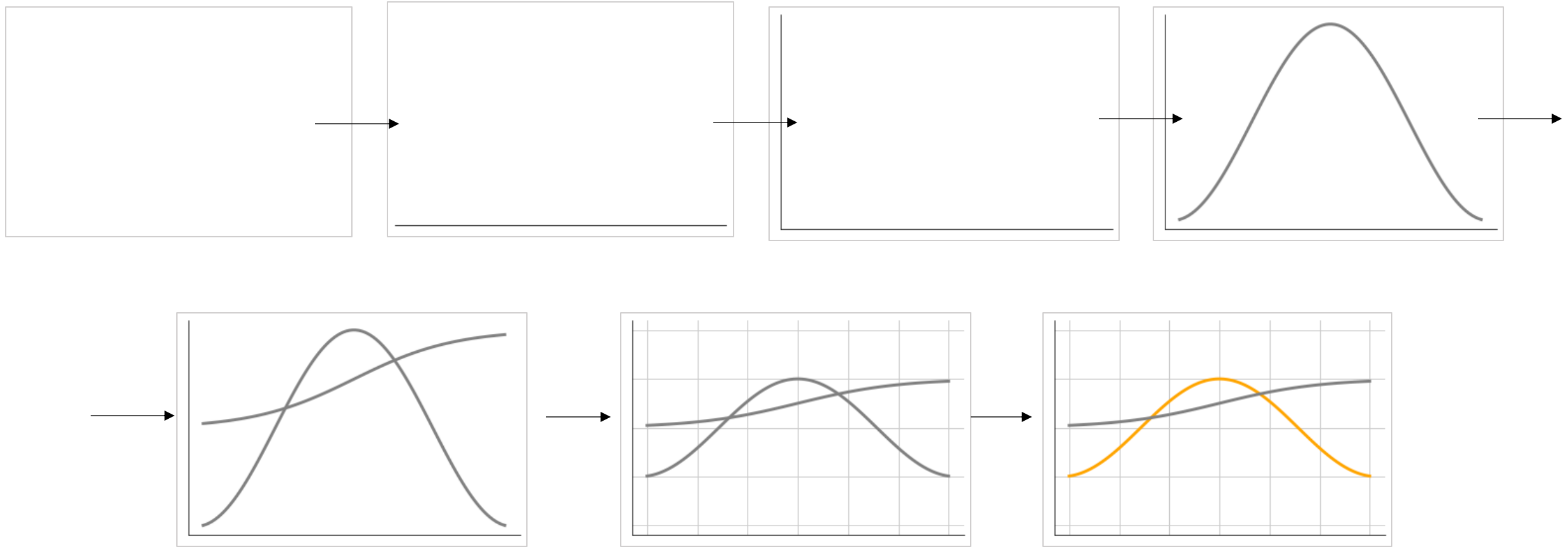
[89] 1 def dynamics(x,u):
2     xnext = x + u
3     return xnext % 2

[90] 1 # Solve the optimal control problem!
2 def optimal_designer(x0, T):
3     x = x0
4     xtraj = []
5     xtraj.append(x)
6
7     for t in range(T):
8         uopt = np.zeros(len(x)) # zero control
9         for u in get_actions(x):
10            xnext = dynamics(x,u)
11            r = reward(xnext)
12            if r > reward(dynamics(x,uopt)):
13                uopt = u
14            # execute optimal control
15            x = dynamics(x,uopt)
16            xtraj.append(x)
17
18     return xtraj
```

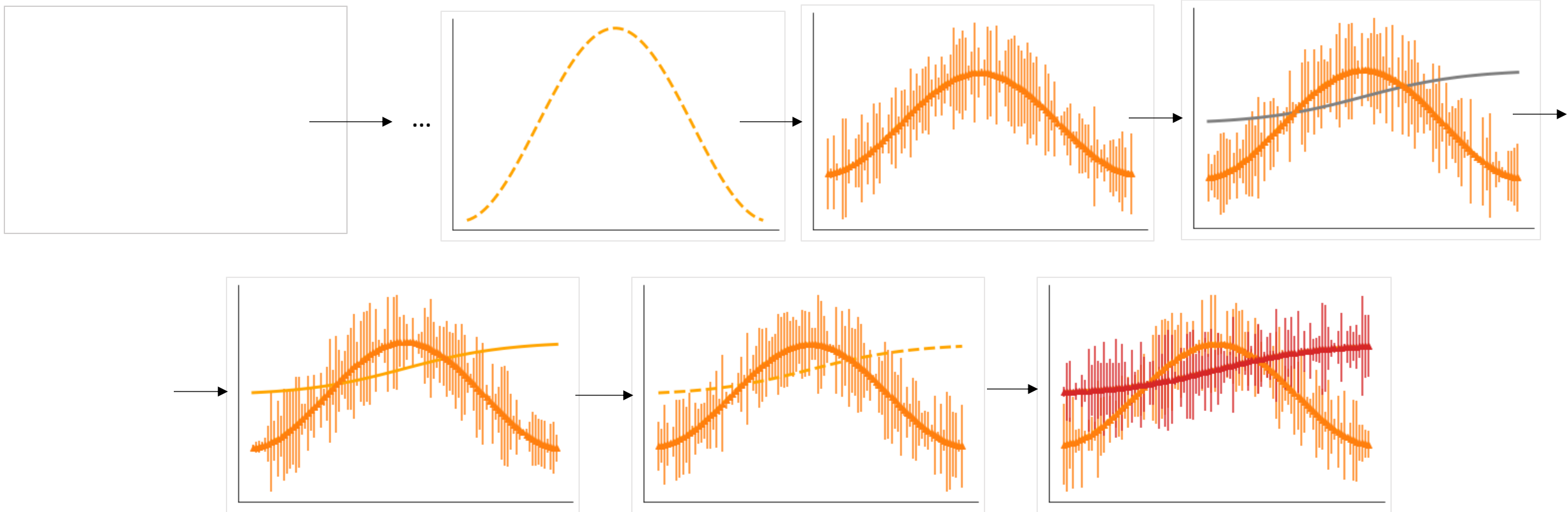
*The Optimal Design!*



*The Optimal Design Trajectory:  $\text{Info}(x^T)$  –  $\text{ViewerEffort}(x^T)$*

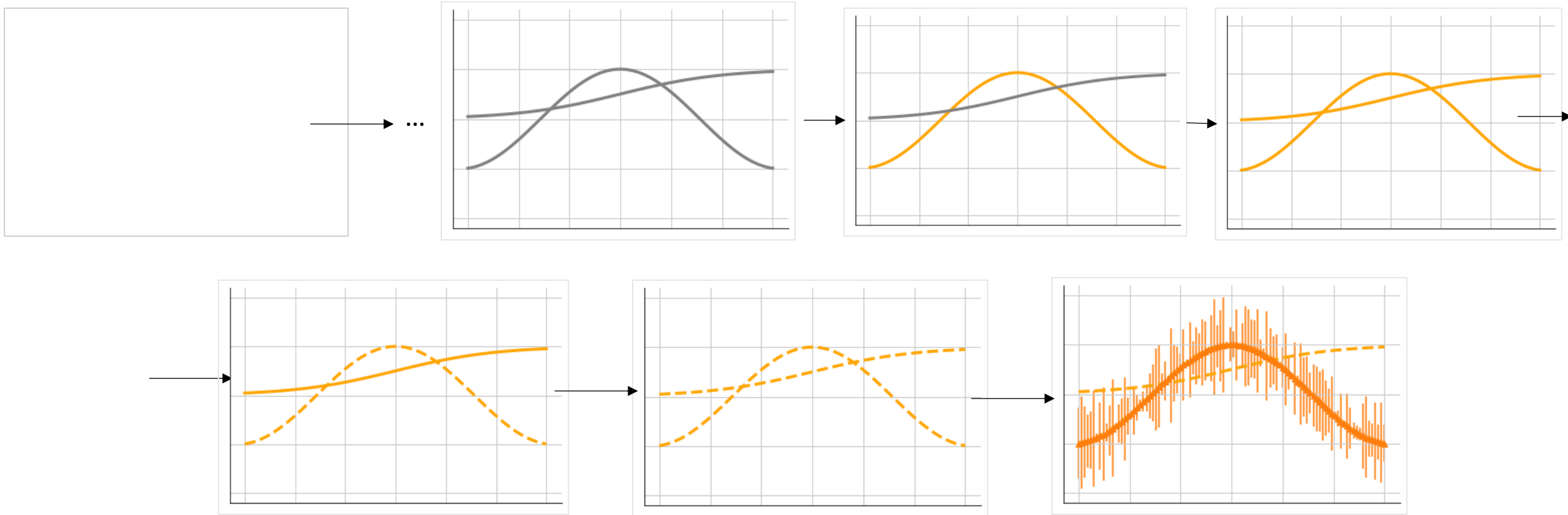


If you only care about  $\text{Info}(x^T)$ ...





If you care more about Info:  $100 \cdot \text{Info}(x^T) - \text{ViewerEffort}(x^T) \dots$



# *Front Figures*

$$\max_{u^0} \text{Info}(x^1) - \text{ViewerEffort}(x^1)$$

$$s. t. \quad \begin{aligned} x^1 &= f(x^0, u^0) \\ x^0 &= x_{init} \end{aligned}$$

*Communicate your contribution.  
Show what makes your work “impactful”?*

$$\max_{u^0} \text{Info}(x^1) - \text{ViewerEffort}(x^1) + \text{Impact}(x^1)$$

$$s. t. \quad \begin{aligned} x^1 &= f(x^0, u^0) \\ x^0 &= x_{init} \end{aligned}$$



# Data Augmentation for Manipulation

Peter Mitrano  
University of Michigan  
Email: pmitrano@umich.edu

Dmitry Berenson  
University of Michigan  
Email: dmitryb@umich.edu

*Abstract*—The success of deep learning depends heavily on the availability of large datasets, but in robotic manipulation there are many learning problems for which such datasets do not exist. Collecting these datasets is time-consuming and expensive, and therefore learning from small datasets is an important open problem. Within computer vision, a common approach to a lack of data is *data augmentation*. Data augmentation is the process of creating additional training examples by modifying existing ones. However, because the types of tasks and data differ, the methods used in computer vision cannot be easily adapted to manipulation. Therefore, we propose a data augmentation method for robotic manipulation. We argue that augmentations should be valid, relevant, and diverse. We use these principles to formalize augmentation as an optimization problem, with the objective function derived from physics and knowledge of the manipulation domain. This method applies rigid body transformations to trajectories of geometric state and action data. We test our method in two scenarios: 1) learning the dynamics of planar pushing of rigid cylinders, and 2) learning a constraint checker for rope manipulation. These two scenarios have different data and label types, yet in both scenarios, training on our augmented data significantly improves performance on downstream tasks. We also show how our augmentation method can be used on data to enable more data-efficient online learning.



Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.

# Data Augmentation for Manipulation

Peter Mitrano  
University of Michigan  
Email: pmitrano@umich.edu

Dmitry Berenson  
University of Michigan  
Email: dmitryb@umich.edu

*Abstract*—The success of deep learning depends heavily on the availability of large datasets, but in robotic manipulation there are many learning problems for which such datasets do not exist. Collecting these datasets is time-consuming and expensive, and therefore learning from small datasets is an important open problem. Within computer vision, a common approach to a lack of data is *data augmentation*. Data augmentation is the process of creating additional training examples by modifying existing ones. However, because the types of tasks and data differ, the methods used in computer vision cannot be easily adapted to manipulation. Therefore, we propose a data augmentation method for robotic manipulation. We argue that augmentations should be valid, relevant, and diverse. We use these principles to formalize augmentation as an optimization problem, with the objective function derived from physics and knowledge of the manipulation domain. This method applies rigid body transformations to trajectories of geometric state and action data. We test our method in two scenarios: 1) learning the dynamics of planar pushing of rigid cylinders, and 2) learning a constraint checker for rope manipulation. These two scenarios have different data and label types, yet in both scenarios, training on our augmented data significantly improves performance on downstream tasks. We also show how our augmentation method can be used on real-robot data to enable more data-efficient online learning.

## I. INTRODUCTION

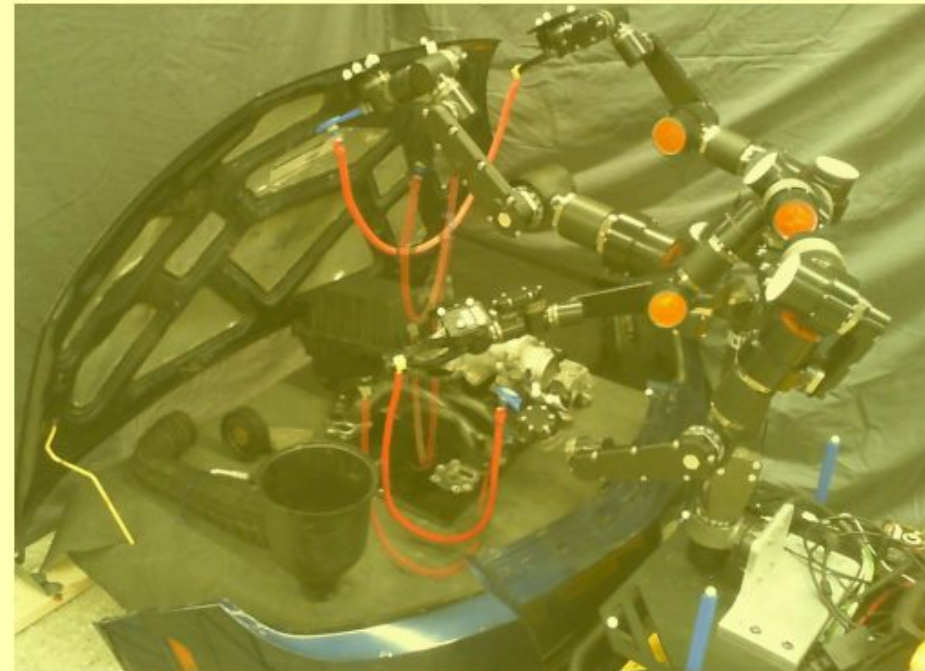


Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.


model [19, 15, 2]. This approach assumes a useful generative



# Data Augmentation for Manipulation

Peter Mitrano  
University of Michigan  
Email: pmitrano@umich.edu

Dmitry Berenson  
University of Michigan  
Email: dmitryb@umich.edu



**Abstract**—The success of deep learning depends heavily on the availability of large datasets, but in robotic manipulation there are many learning problems for which such datasets do not exist. Collecting these datasets is time-consuming and expensive, and therefore learning from small datasets is an important open problem. Within computer vision, a common approach to a lack of data is *data augmentation*. Data augmentation is the process of creating additional training examples by modifying existing ones. However, because the types of tasks and data differ, the methods used in computer vision cannot be easily adapted to manipulation. Therefore, we propose a data augmentation method for robotic manipulation. We argue that augmentations should be valid, relevant, and diverse. We use these principles to formalize augmentation as an optimization problem, with the objective function derived from physics and knowledge of the manipulation domain. This method applies rigid body transformations to trajectories of geometric state and action data. We test our method in two scenarios: 1) learning the dynamics of planar pushing of rigid cylinders, and 2) learning a constraint checker for rope manipulation. These two scenarios have different data and label types, yet in both scenarios, training on our augmented data significantly improves performance on downstream tasks. We also show how our augmentation method can be used on real-robot data to enable more data-efficient online learning.

## I. INTRODUCTION



Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.

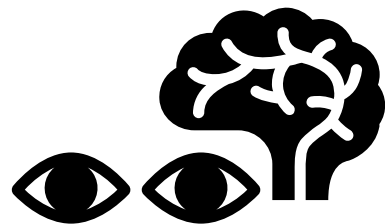
model [19, 15, 2]. This approach assumes a useful generative

# Data Augmentation for Manipulation

Peter Mitrano  
University of Michigan  
Email: pmitrano@umich.edu

Dmitry Berenson  
University of Michigan  
Email: dmitryb@umich.edu

*Abstract*—The success of deep learning depends heavily on the availability of large datasets, but in robotic manipulation there are many learning problems for which such datasets do not exist. Collecting them is often expensive and time-consuming, and therefore learning from data is often a difficult problem. Within computer vision, one common method for increasing the amount of data is *data augmentation*, which involves creating additional data by applying transformations to existing data. However, because these transformations are often not used in computer vision, they can be difficult to apply. Therefore, we propose a method for data augmentation in robotic manipulation. We use a set of principles that are relevant, and diverse, to generate data. We use these principles to formalize data augmentation as an optimization problem, with the objective function derived from physics and knowledge of the manipulation domain. This method applies rigid body transformations to trajectories of geometric state and action data. We test our method in two scenarios: 1) learning the dynamics of planar pushing of rigid cylinders, and 2) learning a constraint checker for rope manipulation. These two scenarios have different data and label types, yet in both scenarios, training on our augmented data significantly improves performance on downstream tasks. We also show how our augmentation method can be used on real-robot data to enable more data-efficient online learning.



*Brain expects a relationship between **title**, **abstract**, **front fig***

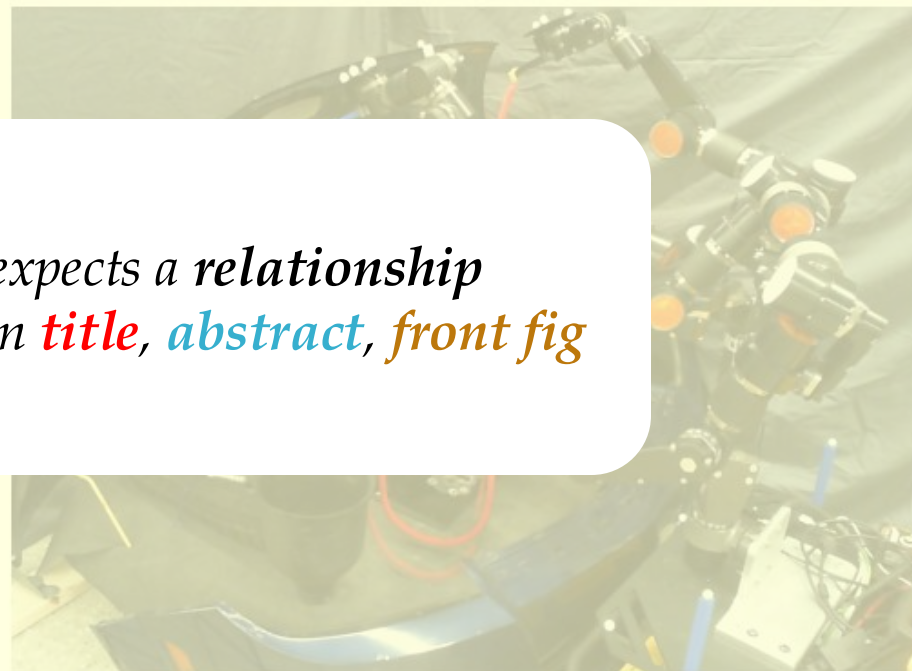


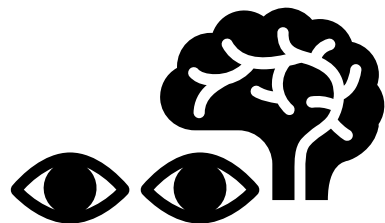
Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.



# Data Augmentation for Manipulation

Peter Mitrano  
University of Michigan  
Email: pmitrano@umich.edu

Dmitry Berenson  
University of Michigan  
Email: dmitryb@umich.edu



$$\max_{u^0} \text{Info}(x^1) - \text{ViewerEffort}(x^1) + \text{Impact}(x^1)$$

*Impact is increased if  
the relationship is clear*

domain. This method applies rigid body transformations to trajectories of geometric state and action data. We test our method in two scenarios: 1) learning the dynamics of planar pushing of rigid cylinders, and 2) learning a constraint checker for rope manipulation. These two scenarios have different data and label types, yet in both scenarios, training on our augmented data significantly improves performance on downstream tasks. We also show how our augmentation method can be used on real-robot data to enable more data-efficient online learning.

## I. INTRODUCTION



Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.

model [19, 15, 2]. This approach assumes a useful generative

# Data Augmentation for Manipulation

Peter Mitrano  
University of Michigan  
Email: pmitrano@umich.edu

Dmitry Berenson  
University of Michigan  
Email: dmitryb@umich.edu

**Abstract**—The success of deep learning depends heavily on the availability of large datasets, but in robotic manipulation many learning problems for which such datasets do not exist. Collecting these datasets is time-consuming and expensive, and learning from small datasets is an important open problem. Within computer vision, a common approach to a lack of data is data augmentation. Data augmentation is the process of generating additional training examples by modifying existing ones. Because the types of tasks and data differ, the methods from computer vision cannot be easily adapted to manipulation. In this paper, we propose a data augmentation method for robotic manipulation. We argue that augmentations should be valid, diverse, and easy to generate. We use these principles to formalize data augmentation as an optimization problem, with the objective function derived from physics and knowledge of the manipulation task. This method applies rigid body transformations to existing data to create new data. We test our method on two scenarios: 1) learning the dynamics of planar pushing of rigid cylinders, and 2) learning a constraint checker for rope manipulation. These two scenarios have different data and label types, yet in both scenarios, training on our augmented data significantly improves performance on downstream tasks. We also show how our augmentation method can be used on real-robot data to enable more data-efficient online learning.

+ Shows real robot system

- Bg distracts

- No insight into approach

- Disconnect from title



Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.

# Data Augmentation for Manipulation

Peter Mitrano  
University of Michigan  
Email: pmitrano@umich.edu

Dmitry Berenson  
University of Michigan  
Email: dmitryb@umich.edu

**Abstract**—The success of deep learning depends heavily on the availability of large datasets, but in robotic manipulation many learning problems for which such datasets do not exist. Collecting these datasets is time-consuming and expensive, and learning from small datasets is an important open problem. Within computer vision, a common approach to a lack of data is data augmentation. Data augmentation is the process of generating additional training examples by modifying existing ones. Because the types of tasks and data differ, the methods used in computer vision cannot be easily adapted to manipulation. In this paper, we propose a data augmentation method for robotic manipulation. We argue that augmentations should be valid, diverse, and easy to generate. We use these principles to formalize data augmentation as an optimization problem, with the objective function derived from physics and knowledge of the manipulation task. This method applies rigid body transformations to the training data of geometric state and action data. We test our method on two scenarios: 1) learning the dynamics of planar pushing of rigid cylinders, and 2) learning a constraint checker for rope manipulation. These two scenarios have different data and label types, yet in both scenarios, training on our augmented data significantly improves performance on downstream tasks. We also show how our augmentation method can be used on real-robot data to enable more data-efficient online learning.

- + Shows real robot system
- + Lightened by
- No insight into approach
- Disconnect from title

## I. INTRODUCTION

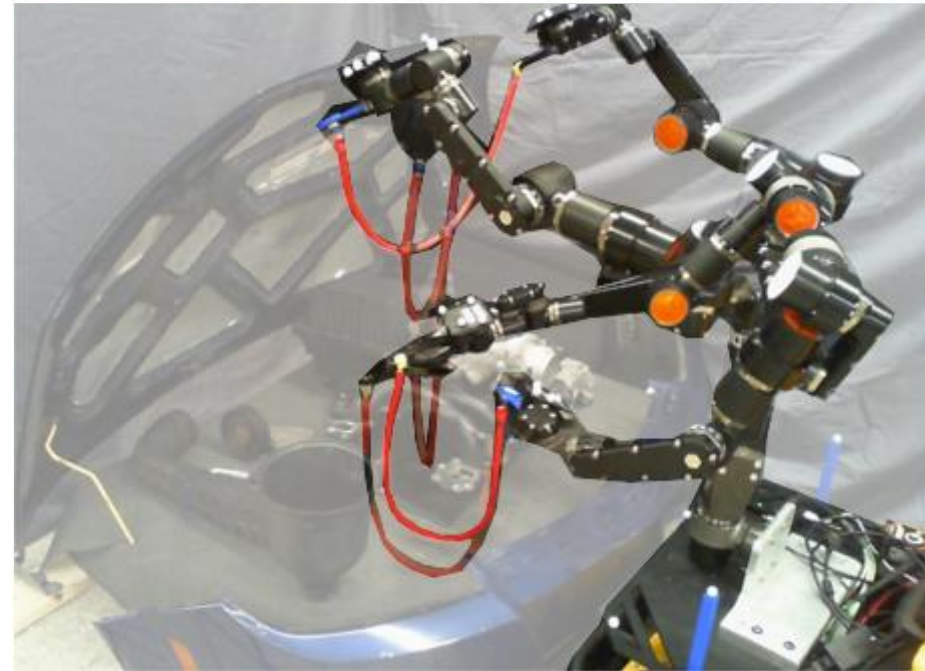


Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.

model [19, 15, 2]. This approach assumes a useful generative

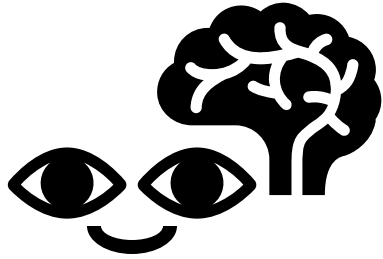
# Data Augmentation for Manipulation

Peter Mitrano  
University of Michigan  
Email: pmitrano@umich.edu

Dmitry Berenson  
University of Michigan  
Email: dmitryb@umich.edu

**Abstract**—The success of deep learning depends heavily on the availability of large datasets, but in robotic manipulation many learning problems for which such datasets do not exist collecting these datasets is time-consuming and expensive, therefore learning from small datasets is an important open problem. Within computer vision, a common approach to a lack of data is data augmentation. Data augmentation is the process of generating additional training examples by modifying existing ones. Because the types of tasks and data differ, the methods from computer vision cannot be easily adapted to manipulation. In this paper, we propose a data augmentation method for robotic manipulation. We argue that augmentations should be valid, useful, and diverse. We use these principles to formalize data augmentation as an optimization problem, with the objective of maximizing the information gained from physics and knowledge of the manipulation method. Our method applies rigid body transformations to geometric state and action data. We test our method on two scenarios: 1) learning the dynamics of planar robot arms with two cylinders, and 2) learning a constraint checker for engine manipulation. These two scenarios have different data requirements, yet in both scenarios, training on our augmented data significantly improves performance on downstream tasks. We show how our augmentation method can be used online to enable more data-efficient online learning.

- + Shows real robot system
- + Lightened by
- + More insight



Relationship between  
title, abstract, front fig

## I. INTRODUCTION

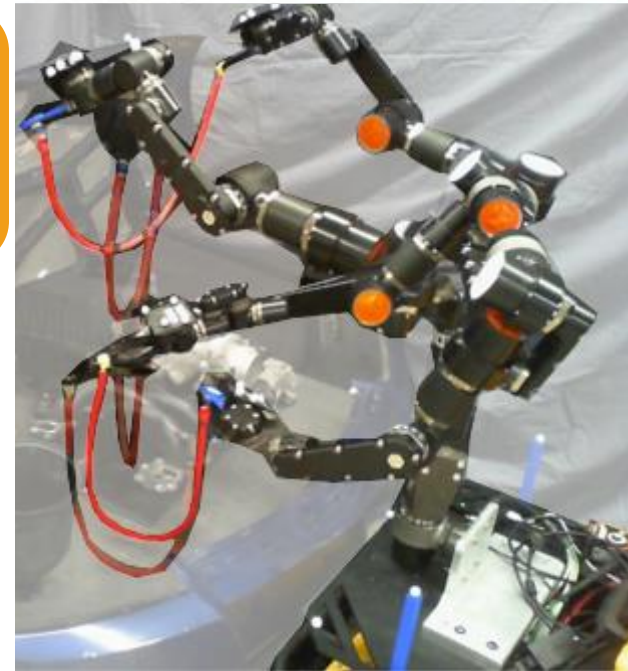
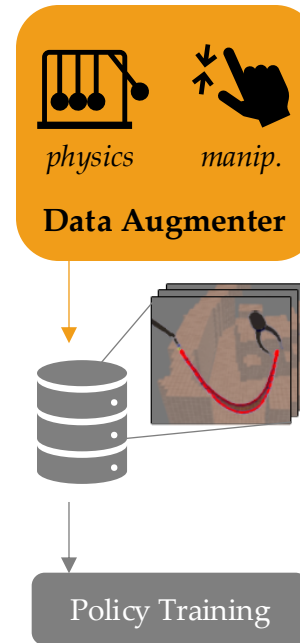


Fig. 1: A mock-up of a car engine bay. The robot must move the rope and place it under the engine without snagging it to set up for lifting the engine. We use data augmentation to improve task success rate during online learning for this task.

model [19, 15, 2]. This approach assumes a useful generative

# A Learning-based Iterative Control Framework for Controlling a Robot Arm with Pneumatic Artificial Muscles

Hao Ma, Dieter Büchler, Bernhard Schölkopf, Michael Muehlebach

In this work, we propose a new learning-based iterative control (IC) framework that enables a complex soft-robotic arm to track trajectories accurately. Compared to traditional iterative learning control (ILC), which operates on a single reference trajectory, we use a deep learning approach to learn across various reference trajectories. The resulting model-based controller computes feedforward actions and is used in a model-free control design. Our method incorporates a model of the system dynamics and by learning about the system dynamics and by learning about the system dynamics, it mitigates the risk of instability. We demonstrate low sample complexity and an excellent tracking performance in real-world experiments. The experiments are conducted on a custom-made robot arm with four degrees of freedom (DoF) actuated with pneumatic artificial muscles. The experiments include high acceleration and high velocity motion.

## I. INTRODUCTION

Pneumatic artificial muscles (PAMs) have been widely used for various tasks, such as aerospace applications, [32], medical applications, [26, 22], and industrial applications, [11, 31]. Due to their high power-to-weight

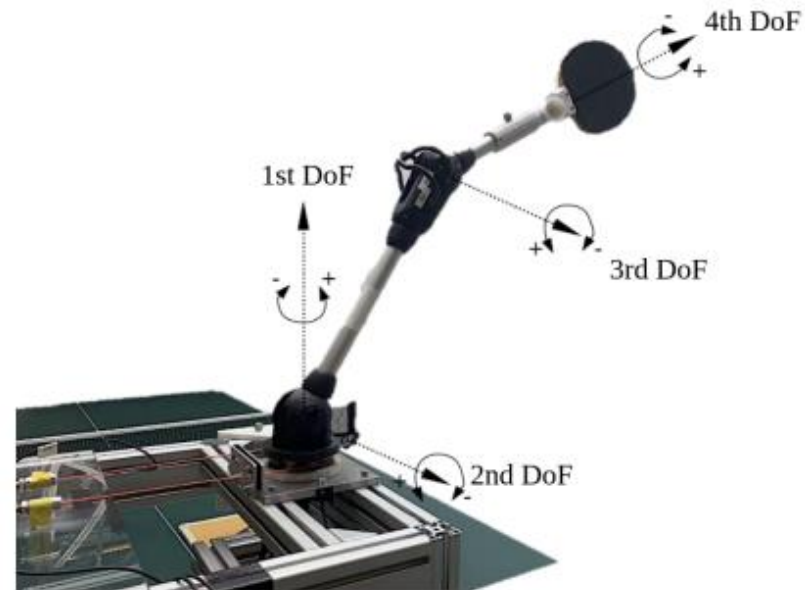


Fig. 1: The figure shows the structure of the robot arm. It has four rotational joints, and each joint is actuated by a pair of PAMs. For simplicity we consider only the first three degrees of freedom in this article. Note that DoF stands for degree of freedom.

- + Shows real robot system
- + Bg removed for clarity
- Irrelevant annotations (?)
- No insight into approach
- Disconnect from title

# Certi fiable Robot Design Optimization using Differentiable Programming

Charles Dawson

Dept. of Aeronautics and Astronautics  
Massachusetts Institute of Technology  
Cambridge, USA  
cbd@mit.edu

Chuchu Fan

Dept. of Aeronautics and Astronautics  
Massachusetts Institute of Technology  
Cambridge, USA  
chuchu@mit.edu

+ Figure shows framework

- Hard to read

- No insight into approach

- Poor image cropping

There is a growing need for computational tools to efficiently design and verify autonomous systems, especially complex robotic systems involving perception, planning, and hardware in the autonomy stack. Differentiable programming has recently emerged as a powerful tool for modeling and optimization. However, very few studies have been done to study how differentiable programming can be used for certifiable end-to-end design optimization. In this paper, we fill this gap by combining differentiable programming for design optimization with a novel statistical framework for certifying the robustness of optimized designs. Our framework enables simultaneous optimization and robustness certification of navigation, perception, planning, control, and hardware subsystems.

Using simulation and hardware experiments, we show how our tool can be used to solve practical problems in robotics. First, we optimize the placement of robot navigation (a design with 10 tunable parameters and 6 tunable parameters) in under 5 minutes to achieve a 0.4X performance improvement compared to the initial

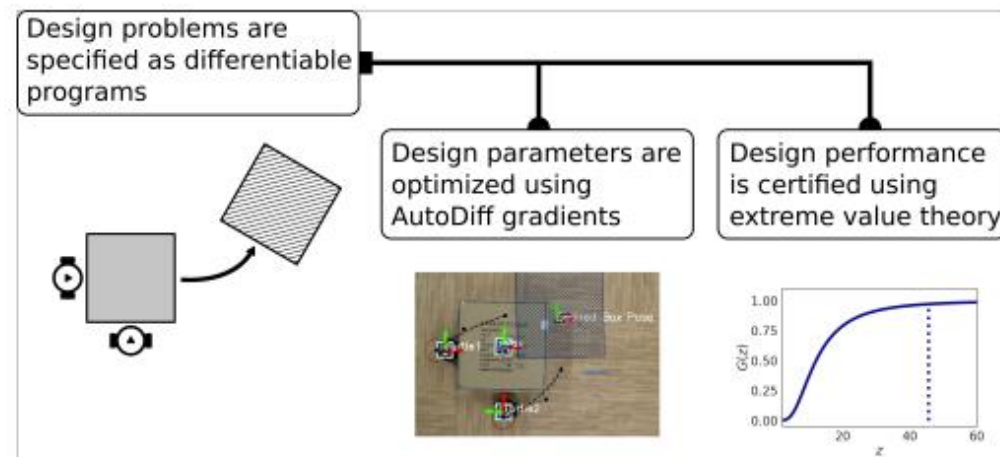


Fig. 1: An overview of our framework for robot design optimization and certification. Differentiable programming allows the user to flexibly specify a robot design problem, which can be efficiently optimized using exact gradients and verified using an extreme value statistical analysis.

# Learning Category-Specific Mesh Reconstruction from Image Collections

Angjoo Kanazawa\*, Shubham Tulsiani\*, Alexei A. Efros, Jitendra Malik

University of California, Berkeley

{kanazawa, shubhtuls, efros, malik}@eecs.berkeley.edu

- + Shows real data / application
- + Visualizes methodology
- + Connected to title

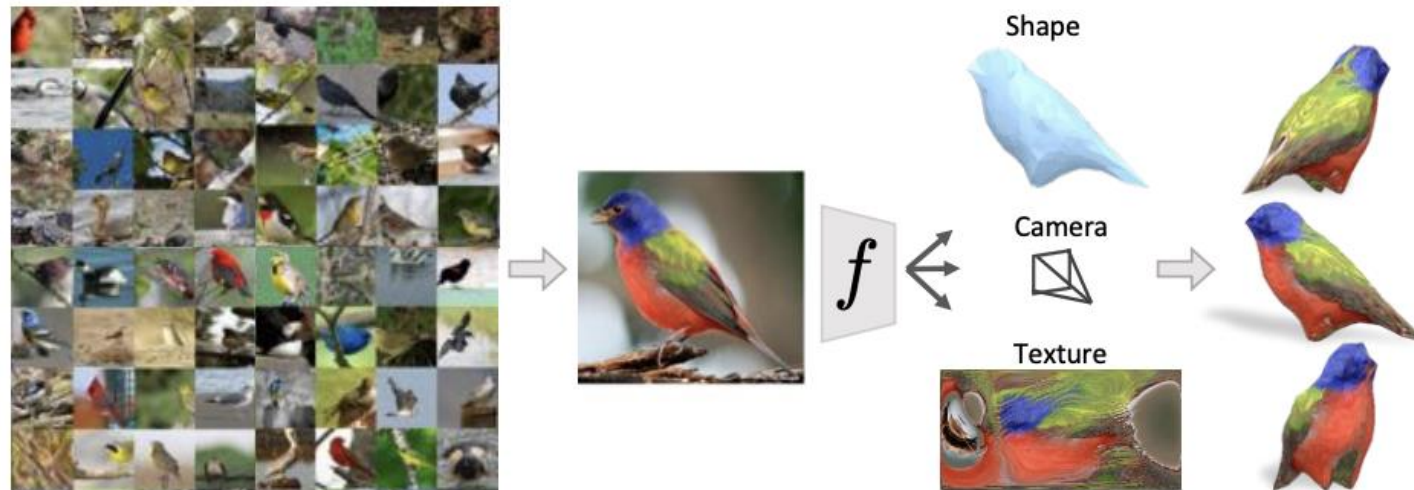


Fig. 1: Given an annotated image collection of an object category, we learn a predictor  $f$  that can map a novel image  $I$  to its 3D shape, camera pose, and texture.

How do we design this reward function?

$$\max_{u^0} \text{Info}(x^1) - \text{ViewerEffort}(x^1) + \text{Impact}(x^1)$$

high Impact



Clear relationship between **title**, **abstract**, **front fig**



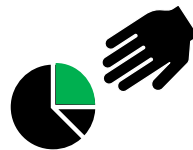
Give **insight** into your approach (e.g., why/how it works)



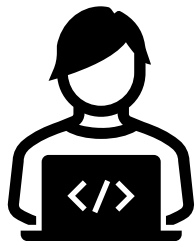
Use **cartoons** for “**framework**” illustrations...



..but also **showcase real robots /data** if you have it



**Highlight** (and/or contextualize) **your contribution**



You  
(the designer)



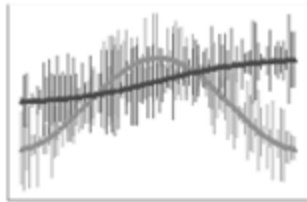
The viewer



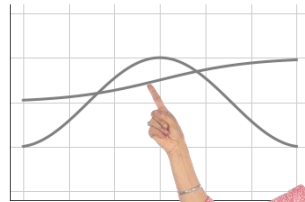
# *From theory to practice: Slides*

*Good news – the same principles apply!*

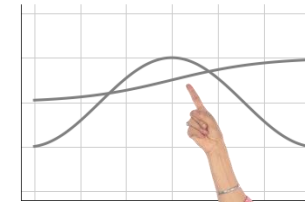
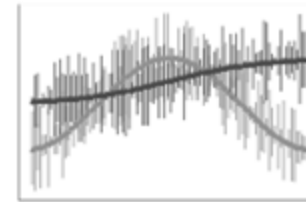
- Increased information
- Increased viewer effort



- Increased information
- Decreased viewer effort



*focus on slide*



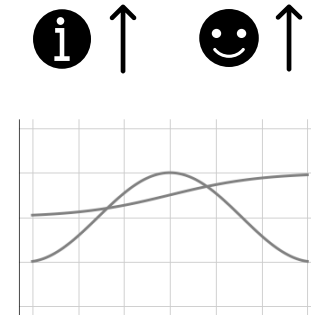
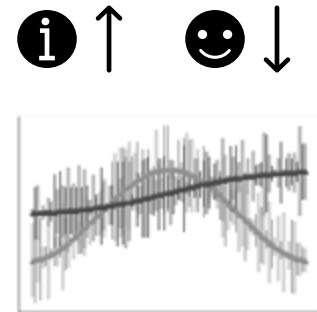
*focus on speaker*

*icons > text*

*If I showed you this slide sequence...*

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

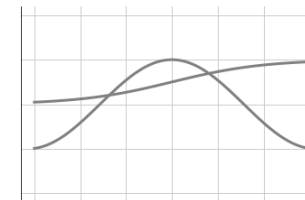
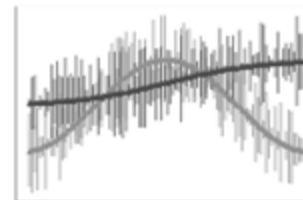
$$\begin{aligned} \text{s. t.} \quad & x^{t+1} = f(x^t, u^t) \\ & x^0 = x_{init} \end{aligned}$$



$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

$$\text{s. t. } \begin{aligned} x^{t+1} &= f(x^t, u^t) \\ x^0 &= x_{init} \end{aligned}$$

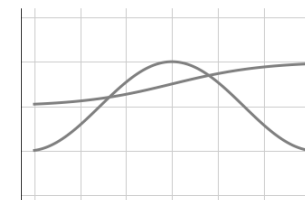
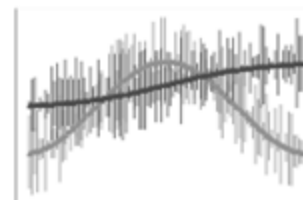
i ↑ ☺ ↓      i ↑ ☺ ↑



$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

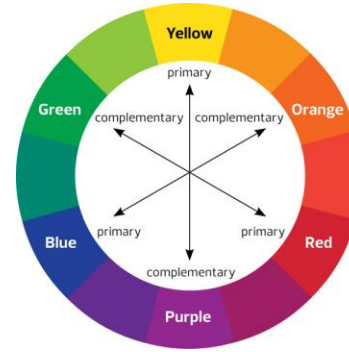
$$\text{s. t. } \begin{aligned} x^{t+1} &= f(x^t, u^t) \\ x^0 &= x_{init} \end{aligned}$$

i ↑ ☺ ↓      i ↑ ☺ ↑



*use colors to encode meaning*

*color "rules"*



*start w/ primary color*

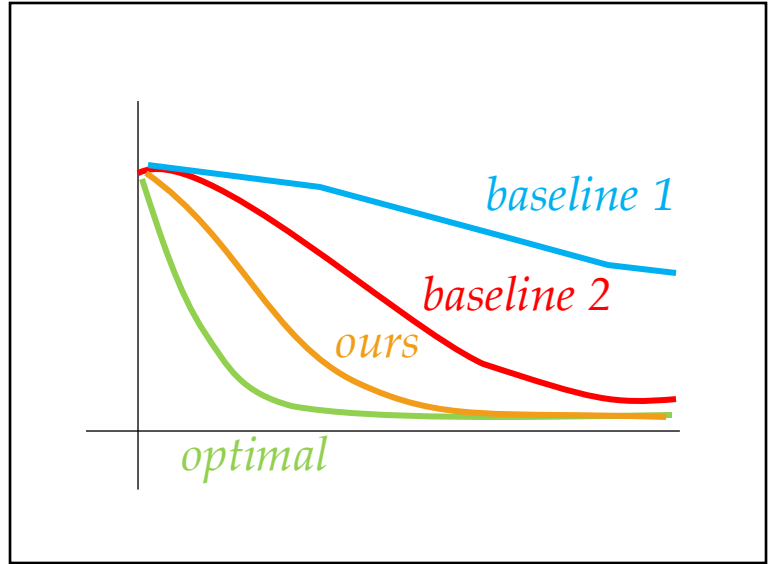
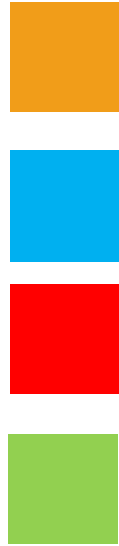
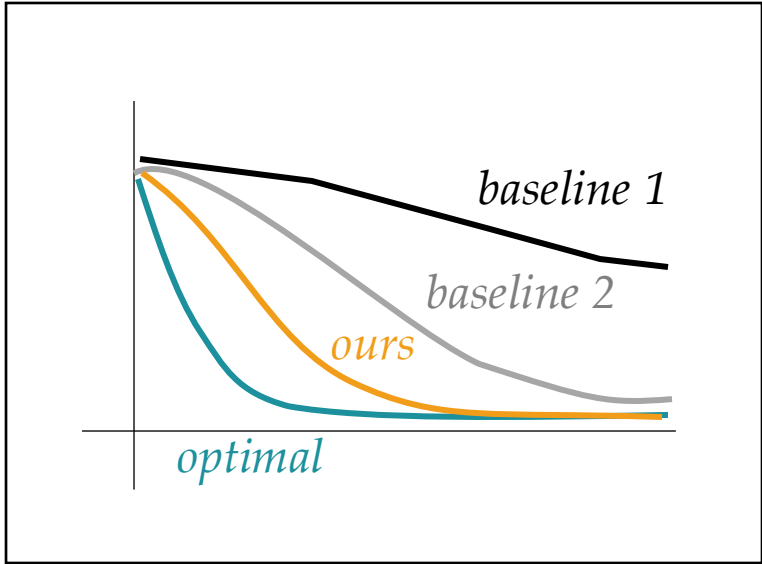


*neutral colors*



*accent color*





$$\begin{aligned} \max_{u^{0:T}} \quad & \text{Info}(x^T) - \text{ViewerEffort}(x^T) \\ \text{s.t.} \quad & x^{t+1} = f(x^t, u^t) \\ & x^0 = x_{init} \end{aligned}$$

*less visible colors*

$$\begin{aligned} \max_{u^{0:T}} \quad & \text{Info}(x^T) - \text{ViewerEffort}(x^T) \\ \text{s.t.} \quad & x^{t+1} = f(x^t, u^t) \\ & x^0 = x_{init} \end{aligned}$$

*more visible colors\**

*\*subject to projector / TV quality and lighting conditions*

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

$$\text{s.t.} \quad x^{t+1} = f(x^t, u^t)$$

$$x^0 = x_{init}$$



*focus on slide*

$$\max_{u^{0:T}} \text{Info}(x^T) - \text{ViewerEffort}(x^T)$$

$$\text{s.t.} \quad x^{t+1} = f(x^t, u^t)$$

$$x^0 = x_{init}$$



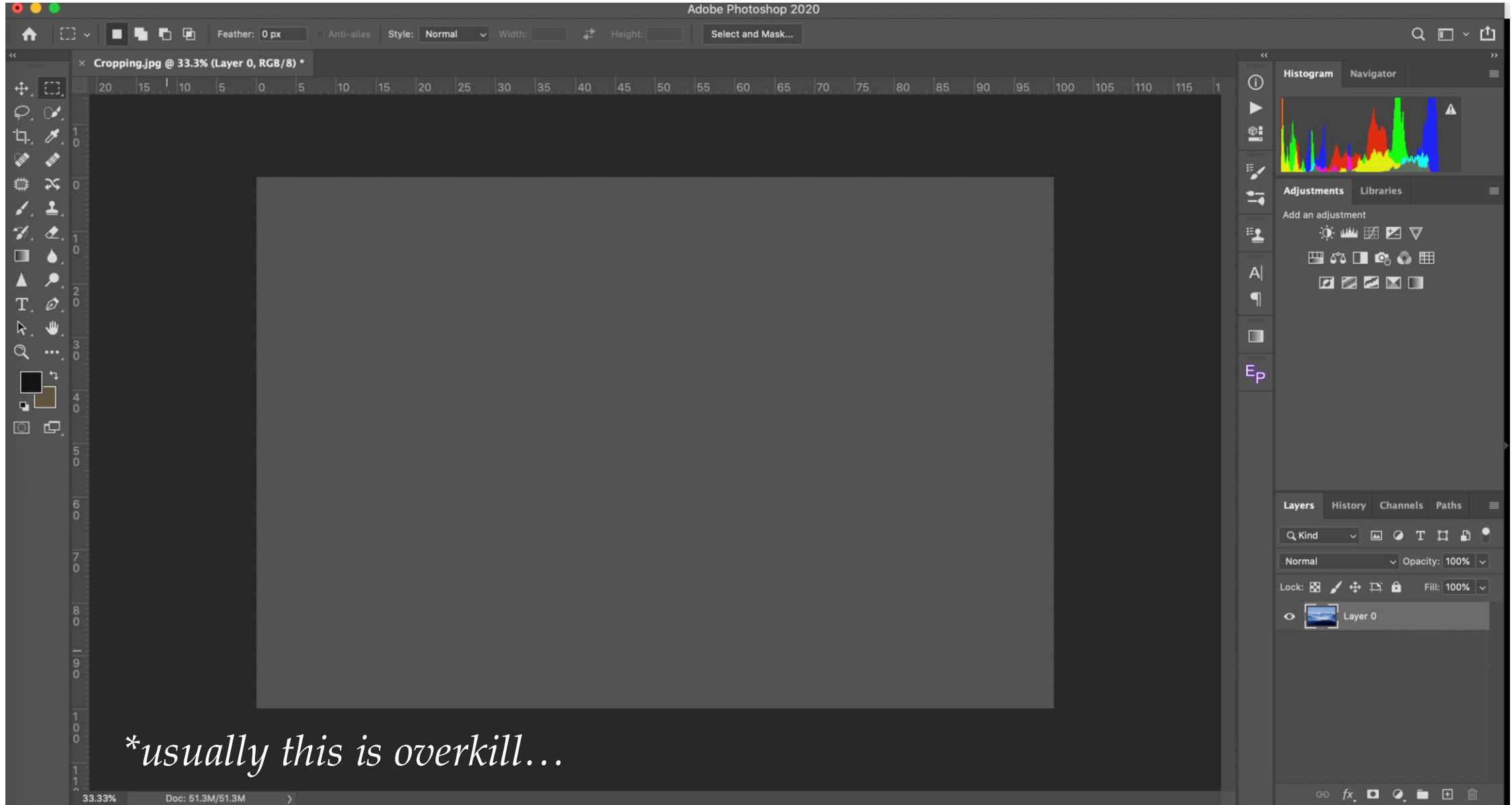
*focus on speaker*



*Where to get* {  
*tools*  
*figures*  
*fonts*  
*color schemes*

Photoshop\*

Tools



*\*usually this is overkill...*

## Free Web-browser-based "Photoshop"

The screenshot shows the Photopea web application interface in a browser window. The address bar shows the URL [photopea.com](https://photopea.com). The browser's address bar includes navigation icons (back, forward, refresh) and extension icons (ABP, etc.).

The application interface features a dark grey header with a menu: File, Edit, Image, Layer, Select, Filter, View, Window, More, and Account (highlighted in red). On the right side of the header, there are links for About, Report a bug, Learn, Blog, and API, along with social media icons for GitHub, Twitter, and Facebook. A notification "Ad blocking detected" is visible on the right.

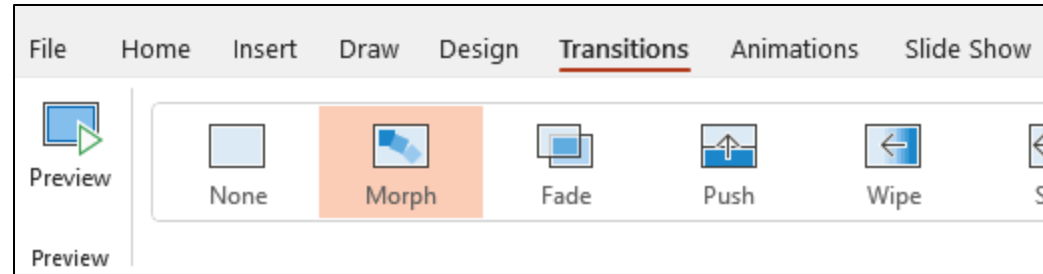
The main content area has a dark background. On the left is a sidebar with navigation options: Home (selected), This Device, Google Drive, Shared Drives, Private Folder, OneDrive, and Dropbox. The center of the page displays the Photopea logo (a teal square with a white swirl) and the word "Photopea" in large white text. Below the logo are three buttons: "★ New Project", "📁 Open From Computer", and "📄 PSD Templates". A large, light grey rectangular area in the center contains the text "Drop any files here".

At the bottom, there is a row of icons representing supported file formats: .PSD (blue square), .AI (orange square), .XD (pink square), .sketch (yellow diamond), .PDF (red square), .XCF (grey cat icon), RAW (grey camera icon), and ANY (black square with white text listing supported formats: JPG, PNG, GIF, TIFF, SVG, DDS, ...).

On the right side, there is a vertical advertisement for "fiverr." with the text "Give your business the perfect logo." and "Get Yours Custom-Made". Below the text is an image of a brown paper shopping bag with a green leaf logo and the word "Naturella" written on it.

*Don't underestimate the power in PowerPoint 😊*

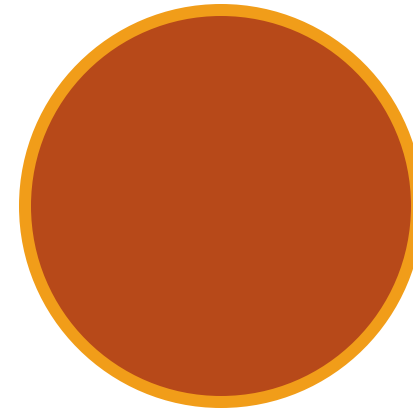
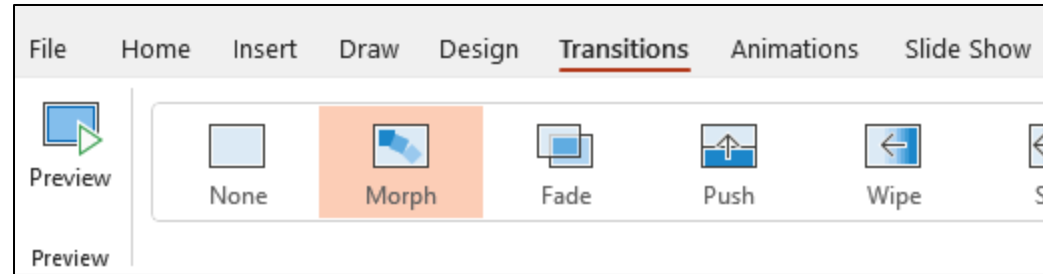
**Morph Transition**  
*makes everything look like you  
spent a lot of time animating it*



# Tools

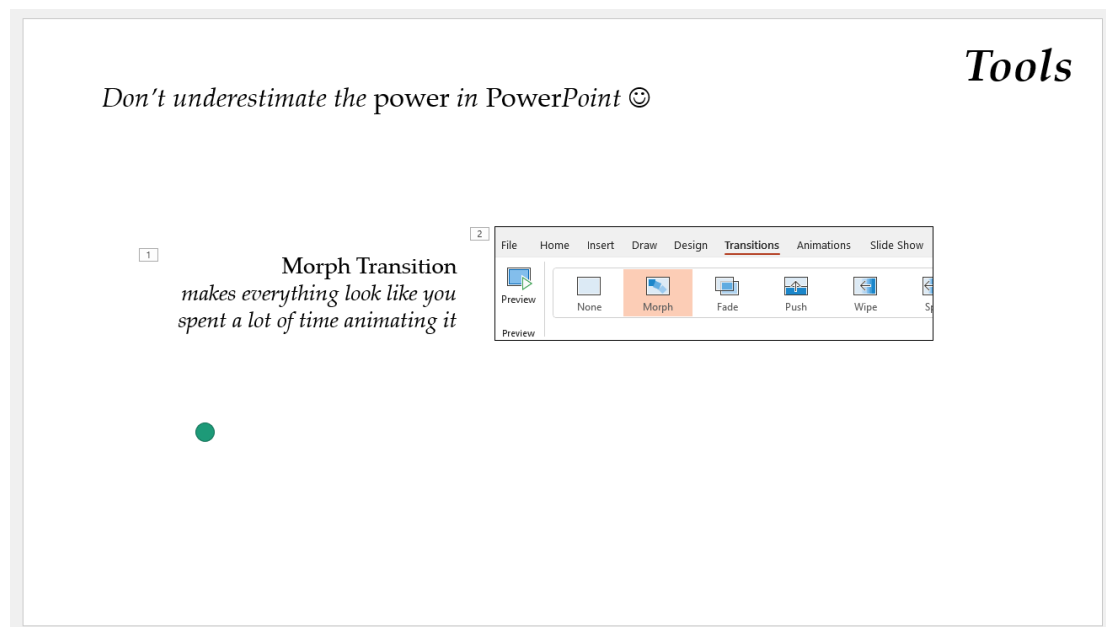
*Don't underestimate the power in PowerPoint 😊*

Morph Transition  
*makes everything look like you  
spent a lot of time animating it*

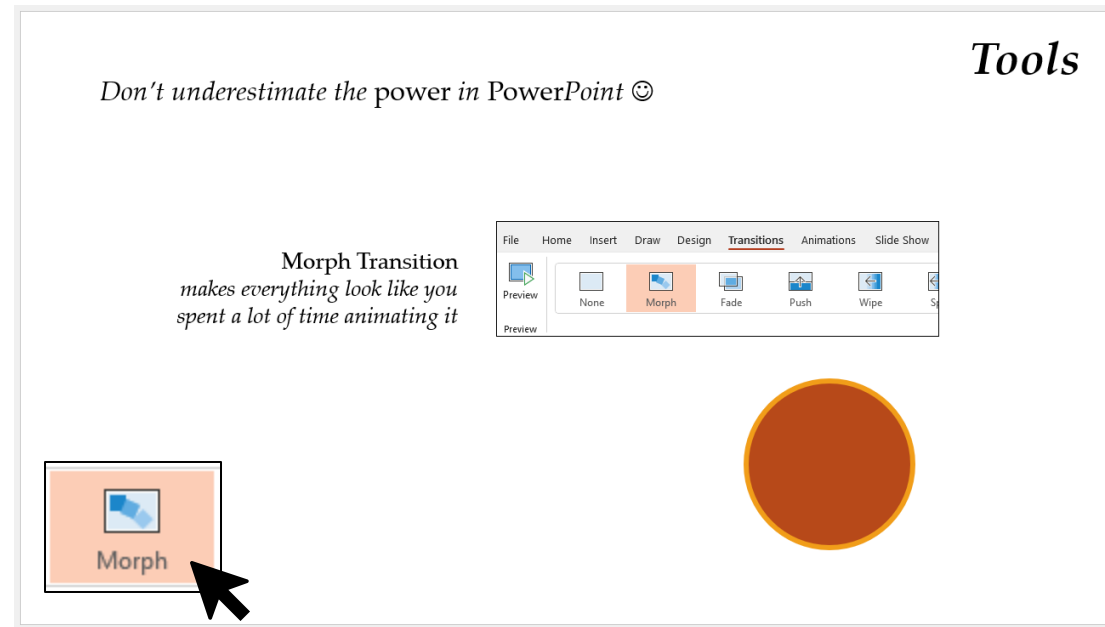


# Tools

*Don't underestimate the power in PowerPoint 😊*



Slide 1



Copy of Slide 1 + change to circle

*Don't underestimate the power in PowerPoint ☺*

Really good practice to animate equations “step by step”

$$\dot{x} = f(x, u)$$

*Don't underestimate the power in PowerPoint ☺*

Really good practice to animate equations “step by step”

$$\dot{x} = f(x, u, d)$$



# *Tools*

*Don't underestimate the power in PowerPoint ☺*

Really good practice to animate equations “step by step”

$$\nabla_x V(x, t) \cdot f(x, u, d)$$

# Tools

*Don't underestimate the power in PowerPoint ☺*

Really good practice to animate equations “step by step”

$$\max_u \min_d \nabla_x V(x, t) \cdot f(x, u, d)$$

# Tools

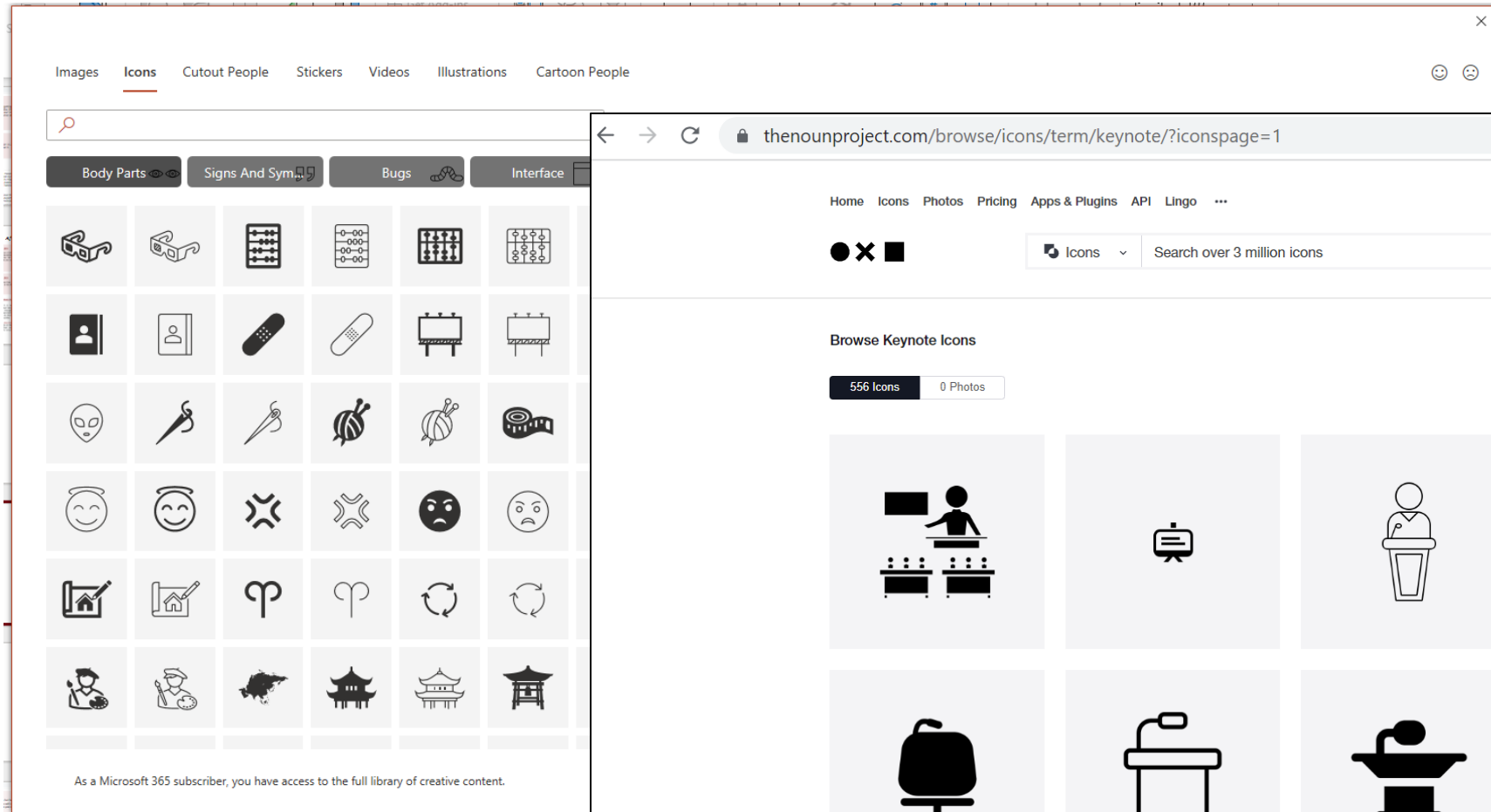
*Don't underestimate the power in PowerPoint ☺*

Really good practice to animate equations “step by step”

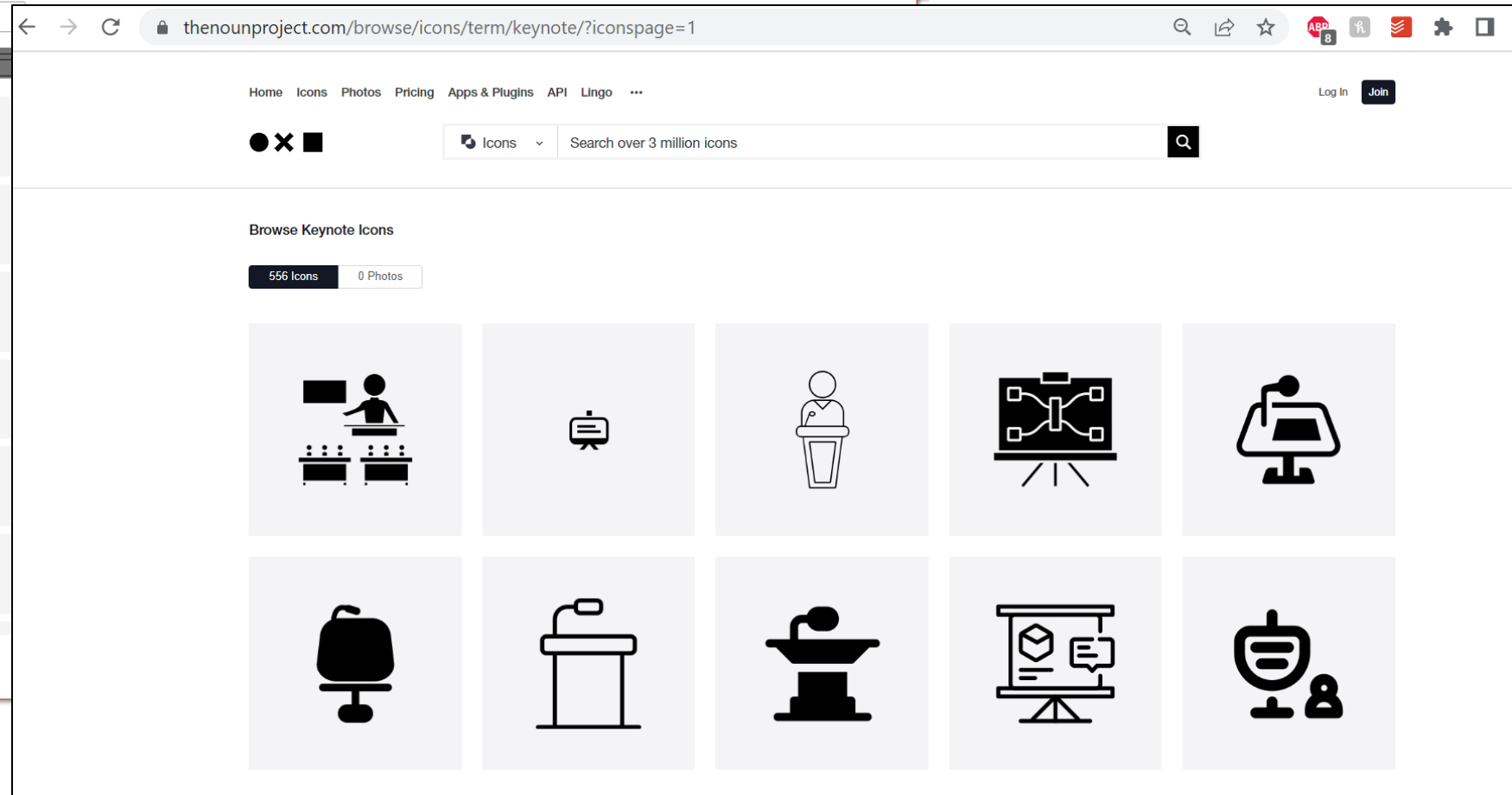
$$\max_u \min_d \nabla_x V(x, t) \cdot f(x, u, d) + \frac{\partial V}{\partial t} = 0$$

# Figures

## PowerPoint Icons



## The Noun Project



# Fonts

## Google Fonts

The screenshot displays the Google Fonts website interface. At the top, the browser address bar shows 'fonts.google.com'. The navigation menu includes 'Fonts', 'Icons', 'Knowledge', and 'FAQ'. A search bar contains the text 'Search fonts', and a preview area shows the sentence 'Sentence Type something' at a size of 40px. Below the search bar are filters for 'Categories', 'Language', and 'Font properties', along with a checkbox for 'Show only variable fonts'. The main content area shows 1424 font families, with results sorted by 'Trending'. Six font family cards are visible, each displaying a preview of the font's appearance using the sentence 'Almost before we knew it, we had left the ground.' (or its Japanese equivalent for Noto Sans Japanese).

Font Family	Designer	Number of Styles	Preview Text
Roboto	Christian Robertson	12 styles	Almost before we knew it, we had left the ground.
Joan	Paolo Biagini	1 style	Almost before we knew it, we had left the ground.
Nuosu SIL	SIL International	1 style	Almost before we knew it, we had left the ground.
Kdam Thmor Pro	Tep Sovichet, Hak Longdey	1 style	Almost before we knew it, we had left the
Open Sans	Steve Matteson	Variable	Almost before we knew it, we had left
Noto Sans Japanese	Google	6 styles	人類社会のすべての構成員の固有の尊厳と平

# Colors

DATA COLOR PICKER powered by LEARN UI DESIGN

UI COURSE UX COURSE BLOG TOOLS DESIGN HACKS

PALETTE SINGLE HUE DIVERGENT

## PALETTE GENERATOR

NUMBER OF COLORS: 3 4 5 6 7 8

BACKGROUND COLOR: LIGHT DARK

#003f5c #58508d #bc5090 #ff6361 #ffa600

*Color palette designers*

DATA COLOR PICKER powered by LEARN UI DESIGN

UI COURSE UX COURSE BLOG TOOLS

PALETTE SINGLE HUE DIVERGENT

## DIVERGENT COLOR SCALE

NUMBER OF COLORS: 5 7 9 11 13




MODIFY MIDPOINT COLOR: BRIGHTNESS, COLOR INTENSITY

BACKGROUND COLOR: LIGHT DARK


#ff9500 #ffb365 #ffd2aa #f1f1f1 #bbdcea #7ec7e2 #00b3db


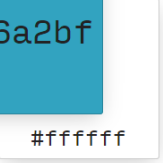
# Colors


**a** ACCESSIBLE COLOR GENERATOR powered by LEARN UI DESIGN


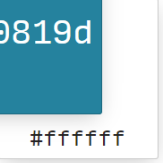
Show me the closest variations of  #4ac4e2  
that contrast against the color  #ffffff  
enough to meet AA Guidelines 

**RESULTS**

**FOR LARGE/BOLD TEXT**    
Try this combo instead:

 #06a2bf  
 #ffffff


**FOR SMALL TEXT**    
Try this combo instead:

 #00819d  
 #ffffff

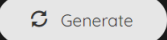
*Deep-learning-generated palettes* 😊

← → ↻ Not secure | colormind.io

Colormind Website Colors Templates Image Upload



40B9B0 🔒 🌐 > 97D2AE 🔒 🌐 < > F3E293 🔒 🌐 < > F1BF61 🔒 🌐 < > FB4135 🔒 🌐 <

 Generate

Colormind is a color scheme generator that uses deep learning. It can learn color styles from photographs, movies, and popular art.

Different datasets are loaded each day, check back tomorrow for even more color inspiration. Visit the [blog](#) for tech info or have a look at our [API](#)

*Accessible colors*

# Academic-Specific Resource

ipb.uni-bonn.de/5min/

## 5 Minutes With Cyrill

The video player shows a man in a white shirt speaking. The video title is "Giving good lectures - 5 Minutes with Cyrill". The video content includes the text "5 Minutes with Cyrill" and "Giving Good Lectures". A red play button is visible in the center of the video frame. In the bottom left corner of the video frame, it says "Watch on YouTube".

Watch on YouTube

+ Giving good lectures - 5 Minutes with Cyrill

The thumbnail shows the same man in a white shirt, gesturing with his hands. The text "5 Minutes with Cyrill" and "Academic Job Interview" is overlaid on the image.

+ Academic Job Interview Talks - 5 Minutes with Cyrill

The thumbnail shows the same man in a white shirt, with his hands clasped. The text "5 Minutes with Cyrill" and "Defending a Proposal" is overlaid on the image.

+ Defending Your Research Project Proposal - 5 Minutes with Cyrill



# *Academic-Specific Resource*



---

# An Optimal Control Approach to Graphic Design

---

**Andrea Bajcsy**  
Carnegie Mellon University  
Pittsburgh, PA 15213  
abajcsy@cmu.edu

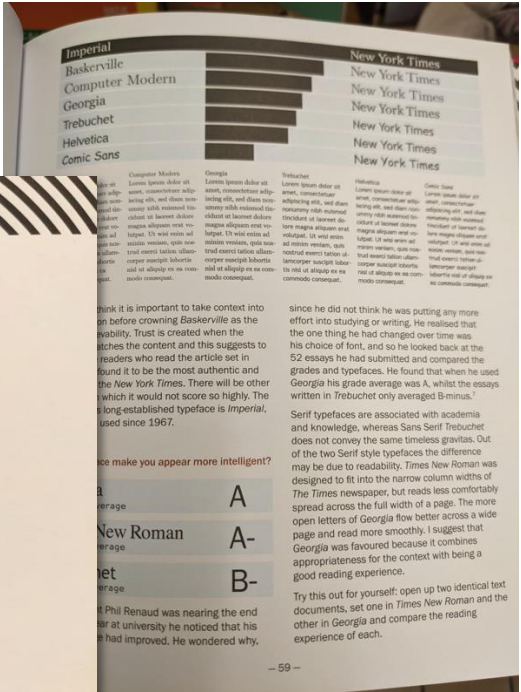
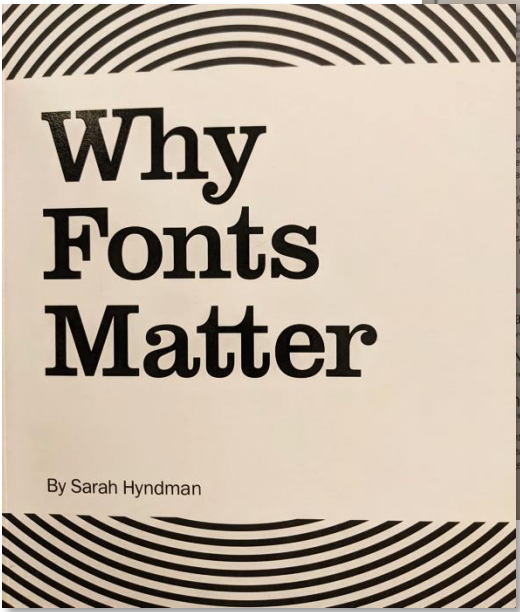
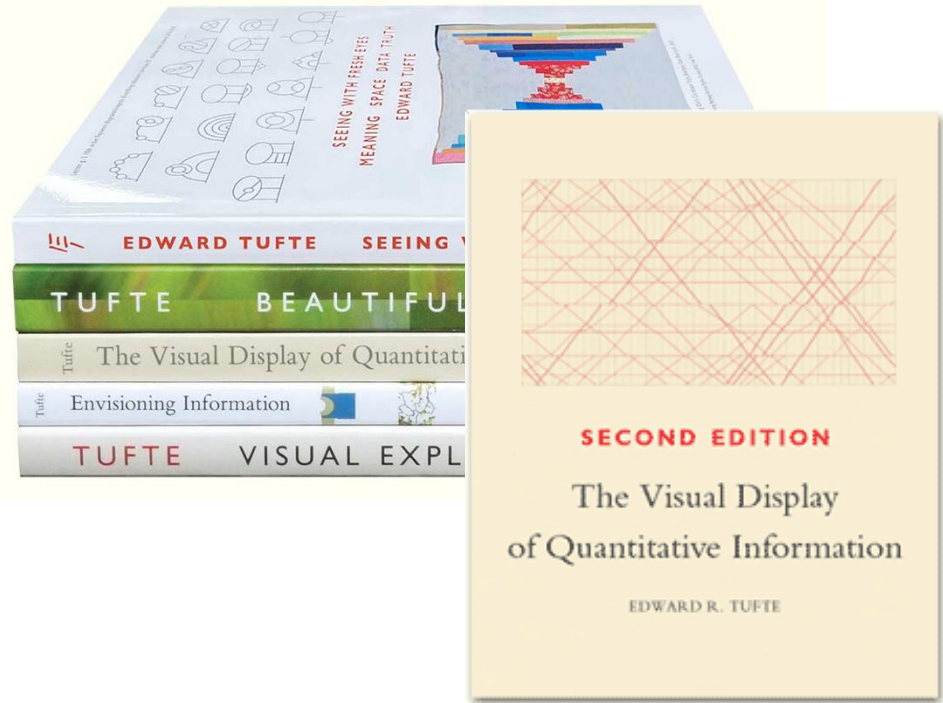
## Abstract

Graphic design challenges are ubiquitous in scientific work: with every new paper researchers must visualize complex data, create technical diagrams, and generate visual aids for talks. Although generative models are revolutionizing the creation of images, technical designs like those mentioned above are still something that experts must create manually. Instead of hoping that scientific figures will emerge from web-trained generative AI, this paper seeks to understand the fundamental *process* behind scientific graphic design. Specifically, we formalize the graphic design process as a multi-objective terminal-cost optimal control problem, trading off information density and viewer effort of the final design. We also present approximation techniques for solving the generally intractable optimal graphic design problem, such as dimensionality reduction, a new algorithm called iterative linearized graphic design (iLGD), and greedy strategies. With this formalism and approximations in hand, we present several exciting frontiers related to preference-based reward learning and generative model alignment with graphic designer behavior.

## 1 Introduction

Modern scientific research increasingly relies on visuals to communicate complex technical results

# Actual graphic design resources



+ many, many more online (esp. YouTube) and in bookstores

# Research Skills

## *Figures & Visuals*

*Think: tradeoff  
between  
information vs.  
viewer effort*

Use heuristics:

- dimensionality reduction
- sketch first; code later
- highlight key idea

Tools:

- Photoshop (e.g. to remove backgrounds)
- [thenounproject.com](http://thenounproject.com) (for icons)
- Color palette generators